

Titre: Approche heuristique pour le placement des machines virtuelles
Title: dans un environnement infonuagique de grande taille

Auteur: Mohamed Mourad Mamlouk
Author:

Date: 2017

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Mamlouk, M. M. (2017). Approche heuristique pour le placement des machines
virtuelles dans un environnement infonuagique de grande taille [Mémoire de
Citation: maîtrise, École Polytechnique de Montréal]. PolyPublie.
<https://publications.polymtl.ca/2716/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/2716/>
PolyPublie URL:

**Directeurs de
recherche:** Samuel Pierre
Advisors:

Programme: Génie informatique
Program:

UNIVERSITÉ DE MONTRÉAL

APPROCHE HEURISTIQUE POUR LE PLACEMENT DES MACHINES VIRTUELLES
DANS UN ENVIRONNEMENT INFONUAGIQUE DE GRANDE TAILLE

MOHAMED MOURAD MAMLOUK

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)

AOÛT 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

APPROCHE HEURISTIQUE POUR LE PLACEMENT DES MACHINES VIRTUELLES
DANS UN ENVIRONNEMENT INFONUAGIQUE DE GRANDE TAILLE

présenté par : MAMLOUK Mohamed Mourad

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

Mme BELLAÏCHE Martine, Ph. D., présidente

M. PIERRE Samuel, Ph. D., membre et directeur de recherche

M. QUINTERO Alejandro, Doctorat, membre

DÉDICACE

A mes parents ...

REMERCIEMENTS

Je tiens à remercier mon directeur de recherche M. Samuel Pierre, pour ses conseils avisés et son écoute qui ont été prépondérants pour la bonne réussite de ce projet.

Je souhaite aussi remercier tous les membres du Laboratoire de recherche en réseautique et informatique mobile principalement Valérie, Saida, Lamia, Marième, Éric et Amirhossein.

Je n'oublierai pas ma famille qui m'a supporté et encouragé tout au long de mes études.

Merci à mes amis qui n'ont soutenu du début jusqu'à la fin.

Merci à tous !

RÉSUMÉ

L'informatique en nuage est un nouveau paradigme qui fait référence à la mise en disposition des ressources d'une infrastructure informatique permettant de présenter un ensemble de services à la demande. Toutefois, un centre de données pourrait participer grandement dans l'augmentation de l'empreinte carbone si le procédé utilisé par le fournisseur de services ne présente pas une bonne gestion des ressources disponibles.

Un fournisseur de services cloud pense principalement aux gains que ses data centers pourraient lui engendrer. Il devra donc trouver un moyen efficace pour que ses bénéfices ne soient jamais diminués. Si de plus, ce dernier pense à la réduction de l'empreinte carbone que pourraient générer ses data centers, il devra trouver un compromis entre les bénéfices qu'ils visent à atteindre et l'empreinte écologique tout en respectant l'accord de niveau de service avec ses clients.

Un moyen pour satisfaire ces compromis serait de revoir la disposition des différents composants à l'intérieur du data center. Les machines virtuelles étant une des variables qui pourraient grandement participer dans la définition des stratégies pour la réduction de l'empreinte carbone, le fournisseur de services se voit dans l'obligation de trouver une solution efficace et peu coûteuse. Ainsi, trouver un bon placement des machines virtuelles dans un environnement infonuagique est une étape importante pour les fournisseurs de services afin d'améliorer l'efficacité énergétique de leurs centres de données.

Les travaux antérieurs élaborés pour la résolution du problème de placement des machines virtuelles ne regroupent souvent pas toutes les informations concernant les composants dans un centre de données. En effet, chaque composant a sa propre consommation d'énergie et participe de près ou de loin à la définition de l'empreinte écologique d'un data center.

Ce mémoire présente le modèle implémenté permettant de calculer l'empreinte carbone d'un centre de données. À travers ce modèle, nous minimisons la fonction objectif à partir des algorithmes de métaheuristique. Nous concevons un algorithme mémétique, une hybridation entre l'algorithme génétique que nous implémentons d'abord seul et l'algorithme de recherche tabou. Nous implémentons ensuite l'algorithme de recuit simulé.

Pour l'évaluation de ces algorithmes, nous comparons les résultats obtenus en elles du point de vue du coût d'empreinte carbone ainsi que le temps d'exécution. De plus, tous les résultats obtenus seront comparés avec l'algorithme de recherche tabou itéré pour diverses tailles de problème.

L'expérimentation menée dans le cadre de ce mémoire montre l'efficacité de l'algorithme mémétique pour la résolution des problèmes de grande taille lors du placement des machines virtuelles dans un environnement infonuagique.

ABSTRACT

Cloud computing is a new paradigm that refers to the provision of resources of an IT infrastructure allowing to present a set of services on demand. However, in this context, a data center could play a significant role in increasing the carbon footprint, if the process used by the service provider does not adequately manage the available resources. The service provider will therefore need to find the best allocation of its data centers components to improve the energy efficiency.

Since virtual machines are one of the components that could play an important role in the definition of strategies for reducing the carbon footprint, the service provider is intended to find an efficient and inexpensive solution. Hence, we can consider that finding an adequate placement of virtual machines in a cloud environment is an important step for service providers. To solve the virtual machines placement problem in the cloud, we need a mathematical model that allow us to compute the carbon footprint while considering all the data center' components.

In this thesis, we present the model implemented to calculate the carbon footprint of a data center. Through this latter, we minimize the objective function based on the metaheuristic algorithms. We design a memetic algorithm, a hybridization between the genetic algorithm and the tabu search algorithm. Afterwards, we implement the simulated annealing algorithm.

For the evaluation of these algorithms, we compare the results obtained in term of carbon footprint cost, as well as the runtime. All the results obtained will be compared with the iterated tabu search algorithm for various problem sizes. The experiments carried out in our work, demonstrate the effectiveness of the memetic algorithm to solve large-scale problems when trying to allocate virtual machines in data centers.

TABLE DES MATIÈRES

DÉDICACE.....	III
REMERCIEMENTS	IV
RÉSUMÉ.....	V
ABSTRACT	VII
TABLE DES MATIÈRES	VIII
LISTE DES TABLEAUX.....	XI
LISTE DES FIGURES.....	XII
LISTE DES SIGLES ET ABRÉVIATIONS	XIII
CHAPITRE 1 INTRODUCTION.....	1
1.1 Définitions et concepts de base	2
1.1.1 Cloud Computing	2
1.1.2 Data center.....	3
1.1.3 Virtualisation.....	3
1.1.4 Machine virtuelle.....	3
1.1.5 Accord de niveau de service.....	4
1.2 Éléments de la problématique	4
1.3 Objectifs de recherche	6
1.4 Plan du mémoire.....	7
CHAPITRE 2 CONSOMMATION D'ÉNERGIE ET PLACEMENT DES MACHINES VIRTUELLES DANS LES DATA CENTERS.....	8
2.1 Consommation d'énergie	8
2.1.1 Consommation des data centers	9

2.1.2	Consommation des serveurs.....	9
2.1.3	Consommation des machines virtuelles	11
2.2	Réduction de l'empreinte carbone	12
2.3	Placement des machines virtuelles	13
2.4	Heuristiques utilisées pour le placement des machines virtuelles.....	21
2.4.1	Algorithme génétique	22
2.4.2	Algorithme mémétique.....	24
2.4.3	Recuit Simulé	25
2.5	Synthèse des problèmes identifiés.....	26
CHAPITRE 3 APPROCHE PROPOSÉE POUR LA RÉOLUTION DU PROBLÈME.....		28
3.1	Modélisation du problème.....	28
3.2	Réduction de l'empreinte carbone	29
3.2.1	Détermination de la température idéale	29
3.2.2	Réduction du trafic réseau	30
3.3	Résolution du problème	31
3.3.1	Génération de la population initiale	31
3.3.2	Algorithme génétique	32
3.3.3	Algorithme mémétique.....	34
3.3.4	Recuit simulé.....	36
CHAPITRE 4 IMPLÉMENTATION ET ANALYSE DES RÉSULTATS		38
4.1	Paramètres d'implémentation.....	38
4.2	Implémentation des algorithmes de métaheuristique	39
4.2.1	Implémentation de l'algorithme génétique	39
4.2.2	Implémentation de l'algorithme mémétique	42

4.2.3	Implémentation du recuit simulé.....	46
4.3	Influences des paramètres de l'environnement	47
4.3.1	Nombre de data centers	48
4.3.2	Nombre de commutateurs	49
4.3.3	Nombre de châssis.....	50
4.3.4	Nombre de serveurs.....	51
4.3.5	Nombre de machines virtuelles	52
4.3.6	Nombre de liaisons entre les machines virtuelles	53
4.4	Évaluation des algorithmes utilisés	54
CHAPITRE 5	CONCLUSION	58
5.1	Synthèse des travaux réalisés	58
5.2	Limitation des travaux.....	59
5.3	Travaux futurs	60
BIBLIOGRAPHIE	61

LISTE DES TABLEAUX

Tableau 4.1 Caractéristiques des composants de l'environnement	38
Tableau 4.2 Caractéristiques des VMs	39
Tableau 4.3 Instances utilisées pour la définition du paramètre de l'AG	40
Tableau 4.4 Résultats détaillés de l'algorithme génétique.....	42
Tableau 4.5 Instances utilisées pour le choix de la taille de la liste taboue	43
Tableau 4.6 Résultats détaillés de l'algorithme mémétique	45
Tableau 4.7 Résultats détaillés du recuit simulé	47
Tableau 4.8 Comparaison entre la méthode exacte, ITS, GA, MA et SA pour les instances de petite taille.....	55
Tableau 4.9: Comparaison entre ITS, GA, MA et SA pour les instances de grande taille	56
Tableau 4.10 Résultat détaillés du MA pour les instances de grande taille	57

LISTE DES FIGURES

Figure 1.1 Machine virtuelle	4
Figure 2.1 Topologie d'un data center	20
Figure 2.2 Environnement infonuagique.....	21
Figure 2.3 Classification des métaheuristiques	22
Figure 2.4 Représentation du croisement dans l'algorithme génétique	23
Figure 2.5 Représentation de la mutation dans l'algorithme génétique.....	23
Figure 2.6 Acceptation d'une configuration dans le SA.....	25
Figure 3.1 Représentation d'un chromosome	32
Figure 4.1 Choix du paramètre du nombre d'individus de la population initiale	40
Figure 4.2 Comparaison du temps d'exécution entre les trois configurations.....	41
Figure 4.3 Influence de la taille de la liste taboue sur la solution obtenue	44
Figure 4.4 Influence de la variation du nombre de data centers sur l'empreinte carbone	48
Figure 4.5 Influence de la variation du nombre de commutateurs sur l'empreinte carbone	49
Figure 4.6 Influence de la variation du nombre de châssis sur l'empreinte carbone.....	50
Figure 4.7 Influence de la variation du nombre de serveurs sur l'empreinte carbone.....	51
Figure 4.8 Influence de la variation du nombre de machines virtuelles sur l'empreinte carbone ..	53
Figure 4.9 Influence de la variation du flux sur l'empreinte carbone.....	54

LISTE DES SIGLES ET ABRÉVIATIONS

DC	Data center
DCs	Data centers
GA	Genetic Algorithm
ILS	Iterated Local Search
ITS	Iterated Tabu Search
FIFO	First In First Out
MA	Memetic Algorithm
PM	Physical Machine
OS	Operating System
QoS	Quality of Service
SA	Simulated Annealing
SLA	Service-Level Agreement
TS	Tabu Search
VM	Virtual Machine
VMMS	Virtual Machine Management System
VMPP	Virtual Machine Placement Problem
VMs	Virtual Machines

CHAPITRE 1 INTRODUCTION

Dans un monde où l'information ne cesse de croître, nous cherchons à trouver le meilleur moyen pour stocker nos données. Externaliser ces données dans des centres de données ou data centers est une solution qui devrait être étudiée et connaître si cette dernière forme un réel avantage. L'accès à ces données stockées se réalise au moyen de services.

Afin de garantir un accès permanent aux services, un accord de niveau de service est réalisé entre le fournisseur et le client. De son côté, le fournisseur devra satisfaire cet accord sans pour autant en payer les frais. Pour cela, trouver le bon placement des différents composants du data center est devenu la priorité dans toute infrastructure infonuagique.

Les data centers sont de plus en plus considérés comme de grands consommateurs d'énergie du fait de l'augmentation de la demande d'hébergement des services. L'assignation des machines virtuelles aux serveurs au sein de ces mêmes data centers est considérée parmi l'une des tâches les plus cruciales pour le profit en puissance. De ce fait, une configuration optimale doit être réalisée afin de parvenir à minimiser l'énergie et l'empreinte en carbone d'un data center.

Il est bien entendu primordial de tenir compte des exigences du client, car certes c'est important de réduire la puissance énergétique consommée, néanmoins il ne faut absolument pas réduire les performances des services proposées par les data centers.

Plusieurs autres facteurs rendent le processus d'assignation des machines virtuelles aux serveurs des data centers plus complexe puisque, dans certains cas, les machines virtuelles peuvent entrer en interaction avec d'autres machines virtuelles. Ainsi un trafic réseau entre ces deux machines doit être élaboré et donc pris en considération lors de la phase d'assignation.

D'un autre côté, et au cours de l'utilisation des data centers, plusieurs machines virtuelles seront peu ou jamais utilisées et qui occupent en permanence une mémoire dans un serveur qui pourrait ne contenir que peu de machines. Au lieu de les migrer sous un autre serveur, ces machines restent, en permanence, en besoin d'une énergie qui n'est autre qu'une énergie gaspillée. Cette manipulation, selon [1], nécessite une configuration préalable afin de faire une migration automatique et faire reposer des serveurs momentanément.

Plusieurs autres facteurs entrent en jeu pour définir la consommation d'énergie d'un data center. Ces facteurs doivent à tout prix être pris en considération et doivent être gérés de manière efficace.

Dans ce chapitre, nous allons d'abord présenter les définitions et les concepts de base. Ensuite, nous traiterons les éléments de la problématique. Puis nous préciserons les objectifs de recherche pour terminer avec le plan du mémoire.

1.1 Définitions et concepts de base

Cette section sera consacrée à la présentation des différents concepts et définitions nécessaires pour la compréhension du sujet traité. Pour cela, nous présenterons le concept du cloud computing ainsi que les différentes parties qui forment son infrastructure à savoir les data centers, la virtualisation ainsi que les machines virtuelles. Enfin, une définition de l'accord de niveau de service s'impose pour présenter l'obligation qu'un fournisseur de cloud envers ses clients.

1.1.1 Cloud Computing

Le cloud computing ou l'infonuagique est un terme utilisé pour exprimer le fait que tout stockage de données ainsi que leur traitement, se passe au niveau des serveurs situés dans des data centers. L'accès à l'information stockée est fait au travers des services. Pour ce fait, des ressources seront allouées et seront rapidement accessibles et à la demande du client. Dans [2], nous distinguons les différents niveaux de services et modèles de déploiement. Le premier est le logiciel en tant que service. Dans ce cas, le fournisseur de service offre un logiciel. Aucune installation n'est nécessaire pour le client, et ce dernier pourra se connecter de n'importe quel terminal à son application.

Ensuite il y a la plateforme en tant que service. Le fournisseur, ici, met à disposition de son client une plateforme où il pourra développer son application sans se soucier des installations préalables ni de la gestion des utilisateurs.

Enfin il y a l'infrastructure en tant que service. Dans ce cas, le fournisseur de service n'offrira que l'infrastructure allant des data centers jusqu'aux serveurs. Le client devra gérer lui-même l'allocation de ses ressources ainsi que la répartition de ses machines virtuelles.

On peut aussi trouver une autre classification dans le cloud. Cette classification se fait selon les modèles de déploiements. Tout d'abord le cloud privé qui est l'approvisionnement d'une infrastructure complète pour une utilisation limitée à une seule organisation. Ensuite, il y a le cloud

communautaire, qui est, comme son nom l'indique, destiné pour une communauté spécifique de consommateurs. Ce cloud est détenu, géré et exploité par les organisations de la communauté. Puis, il y a le cloud public. Ce type de cloud est destiné pour le grand public et est détenu par une organisation qui gère tous les droits d'accès. Enfin, le cloud hybride qui n'est autre que la combinaison entre le cloud privé et le cloud publique.

1.1.2 Data center

Un data center, ou centre de données, selon [3], est l'élément le plus important pour le stockage ainsi que le traitement des données dans le cloud. En effet, c'est le site physique qui regroupe les composants informatiques tels que les commutateurs, les châssis et les serveurs.

Afin d'assurer un service de qualité, le fournisseur doit passer tout d'abord par une bonne gestion des ressources au sein de son data center. Cette bonne gestion s'assure par le fait que tous les équipements de ce dernier ne se retrouve ni en surutilisation ni en sous-utilisation.

La structure d'un data center est considérée comme complexe dû au fait que le nombre d'équipements électroniques est élevé. Plusieurs armoires électroniques sont présentes au sein de la structure afin d'alimenter les divers commutateurs, châssis et serveurs.

1.1.3 Virtualisation

La virtualisation est une technologie qui consiste à faire abstraire un système d'exploitation invité du matériel sous-jacent. Elle permet l'utilisation des serveurs, systèmes ou application sur un même serveur physique utilisant un système d'exploitation hôte. Sur ce dernier est installé un logiciel de virtualisation, appelé hyperviseur. Celui-ci est responsable de l'émulation du matériel aux OS invités [4].

La virtualisation permet de faire fonctionner un ou plusieurs systèmes d'exploitation invités dans des machines virtuelles au-dessus d'un OS hôte.

1.1.4 Machine virtuelle

Une machine virtuelle est une émulation d'un système informatique capable d'héberger un système d'exploitation. Elle possède les mêmes caractéristiques qu'un ordinateur physique à savoir le CPU, mémoire, disque dur et se comporte exactement comme un système hôte.

Plusieurs logiciels permettent de faire cette abstraction du système d'exploitation invité du matériel sous-jacent. La figure 1.1 illustre l'architecture d'une machine virtuelle.

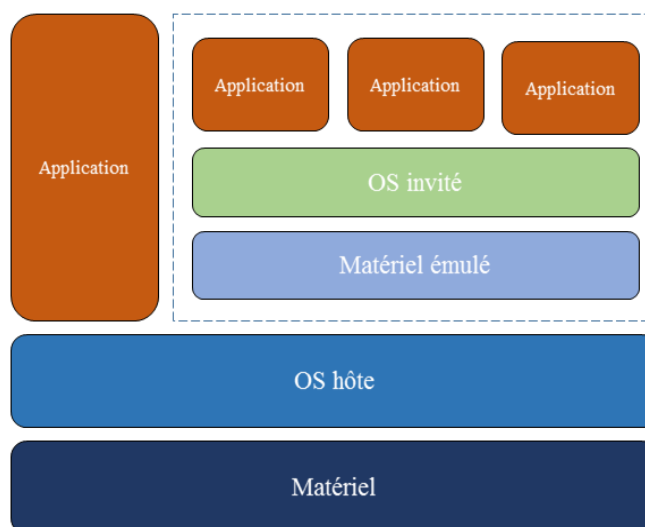


Figure 1.1 Machine virtuelle

1.1.5 Accord de niveau de service

L'accord de niveau de service est un document contenant les obligations du fournisseur de services envers son client. Le SLA permet de garantir que le service proposé soit en parfait accord avec ce que le client demande.

Un SLA contient généralement des informations sur le type des services proposés, ainsi que sur leurs disponibilités. Si un client rencontre des problèmes quant à l'utilisation de ses services, il pourrait suivre les différentes procédures citées selon l'accord de niveau de service. Le fournisseur devrait être en mesure de répondre à ses besoins dans les délais indiqués dans le SLA, sinon il risque de payer une pénalité.[5]

1.2 Éléments de la problématique

Dans un environnement infonuagique, un bon placement de machines virtuelles revient à placer les VMs d'une façon à ce que l'accord de niveau de service soit garanti sans pour autant signaler des pertes au niveau du fournisseur. Au sein d'un data center, il est possible de placer les VMs avec comme principal objectif la réduction de l'empreinte carbone.

L'architecture d'un data center comporte multiples châssis, qui à leurs tours permettent d'accueillir un grand nombre de serveurs dépendamment de leurs tailles. Tous ces composants sont reliés par des commutateurs assurant le trafic réseau.

Placer les VMs au sein d'un data center revient à trouver un espace libre dans les serveurs garantissant le bon fonctionnement des services. Cette tâche dépendra non seulement des caractéristiques de la machine virtuelle en elle-même, mais aussi, des autres VMs avec qui elle est reliée. En effet, séparer des VMs en liaison augmentera le trafic réseau et par conséquent l'empreinte carbone.

Trouver le meilleur placement des machines virtuelles dans un data center est une étape importante qui définit l'empreinte carbone d'un data center donné. Toute VM a des caractéristiques propres à elle, à savoir le processeur, le stockage et la mémoire. Ainsi pour les placer dans les serveurs, il faudra prévoir assez de ressources. De plus, il faudra trouver le meilleur serveur capable d'héberger la machine virtuelle du point de vue trafic réseau.

Tout placement de machines virtuelles dans un data center, nécessite la prise en considération que certaines machines virtuelles doivent être situées dans un même data center voire dans un même serveur. C'est une contrainte qu'un client pourrait parfois demander. Certes l'hétérogénéité des composants complique l'étape de placement des machines virtuelles, néanmoins, il est possible de trouver une planification en un temps moins important que d'essayer toutes les combinaisons possibles de placement.

Pour le placement des machines virtuelles dans les data centers, il faut aussi tenir compte des capacités des serveurs. En effet, ces derniers ne doivent pas atteindre la limite des ressources qu'ils offrent. Dépasser cette limite, les performances des services offerts se verront dégradées surtout que les besoins des machines virtuelles pourraient augmenter au cours du temps. Le nombre de serveurs étant grand, il faudra choisir le serveur adéquat qui participera à long terme à la réduction de la consommation d'énergie. Il faudra aussi tenir compte qu'il faut utiliser le moins possible de serveurs n'appartenant pas au même châssis, car ces derniers ne pourront plus assurer un trafic fluide si les communications s'avèrent trop importantes et par conséquent la température interne augmentera.

Afin de garder la performance demandée, le fournisseur devra en conséquence payer les frais de climatisation. De ce fait, il est préférable de laisser une certaine marge proportionnelle à la capacité maximale du serveur afin de pouvoir gérer tout imprévu.

Dans le but d'éviter toute surprise, le fournisseur sera dans l'obligation de faire une étude de placement des machines virtuelles en fonction des ressources qu'il est capable d'offrir. Augmenter le nombre de machines virtuelles augmentera systématiquement le trafic réseau étant donné que les machines virtuelles généreront des communications inter-cloud.

Plusieurs études se sont intéressées au problème de placement des machines virtuelles dans un data center. Trouver une métaheuristique pour ce problème est une solution peu coûteuse. De ce fait, considérer tous les paramètres dans un même algorithme avec pour objectif la réduction de l'empreinte carbone revient à trouver un modèle complet qui permet de calculer le coût global d'un data center en termes d'empreinte écologique.

Le modèle utilisé devra prendre en considération toutes les liaisons entre les machines virtuelles afin de définir le coût du trafic réseau qui pourra engendrer une forte consommation d'énergie. Aussi, il devra inclure le facteur d'émission des data centers. Étant donné que chacun de ces derniers regroupe plusieurs équipements et qu'il participe à l'augmentation de l'empreinte carbone d'un data center, il est nécessaire de calculer la puissance énergétique de chaque composant ainsi que sa dissipation de puissance.

1.3 Objectifs de recherche

L'objectif principal de ce mémoire est de concevoir une méthode heuristique capable de résoudre le problème de placement des machines virtuelles dans un environnement infonuagique dans l'optique de minimiser l'empreinte carbone. De manière plus spécifique, ce mémoire vise à :

- Implémenter un modèle mathématique capable de mesurer l'empreinte carbone dans un data center et permettant de calculer la consommation d'énergie de tous les composants dans un data center ainsi que le trafic réseau entre les machines virtuelles;
- Concevoir et implémenter une approche heuristique pour la minimisation de l'empreinte carbone au sein d'un data center pour les problèmes de grande taille;

- Évaluer la méthode de résolution en comparant les résultats obtenus avec d'autres approches de résolution ainsi qu'avec la méthode exacte.

1.4 Plan du mémoire

Ce mémoire se compose de cinq chapitres et s'organise comme suit : le premier chapitre étant l'introduction, nous présenterons les différents concepts de base, les éléments de la problématique ainsi que les objectifs de recherche. Le deuxième chapitre sera consacré à la revue de littérature et présentera ainsi les différents travaux réalisés dans le cadre du problème de placement des machines virtuelles dans un environnement infonuagique. Le troisième chapitre décrira la méthodologie utilisée pour la réalisation du problème. Nous détaillerons les différents algorithmes utilisés. Le quatrième chapitre présentera quant à lui les résultats obtenus en analysant les détails et les performances de chaque algorithme. Enfin, le cinquième chapitre sera consacré à la conclusion du mémoire.

CHAPITRE 2 CONSOMMATION D'ÉNERGIE ET PLACEMENT DES MACHINES VIRTUELLES DANS LES DATA CENTERS

Les data centers englobent dans certains cas, un nombre considérable de machines virtuelles. Chacune de ces dernières a des caractéristiques spécifiques à elle telles que la capacité de stockage, la mémoire vive ainsi que le processeur. Pour le placement des VMs dans les serveurs des data centers, il faudra prévoir que toutes les caractéristiques de ces machines virtuelles sont satisfaites afin de respecter l'accord de niveau de service.

L'étape d'assignation des machines virtuelles aux serveurs est une tâche considérée importante pour définir la consommation d'énergie d'un data center. Ainsi, il faudra trouver un compromis entre la qualité des services offerts par les data centers et la consommation d'énergie.

Plusieurs techniques, pour la réduction de la consommation d'énergie dans un data center, seront énumérées. Ces techniques ont été présentées dans d'autres études et permettront de bâtir, sur ces dernières, notre travail à effectuer.

Dans ce chapitre, nous présenterons la consommation d'énergie d'un data center, d'un serveur ainsi que d'une machine virtuelle. Ensuite, nous allons introduire le problème de réduction de l'empreinte carbone. Puis, nous discuterons des précédents travaux connexes qui avaient pour objectif la résolution du problème de placement des VMs pour ainsi présenter les différents algorithmes utilisés dans ce travail. Une synthèse des problèmes identifiés précédemment clôturera ce chapitre.

2.1 Consommation d'énergie

La densité et la capacité de stockage élevée des data centers alourdissent d'une manière importante la facture énergétique sur le plan économique et également écologique. De plus, afin de pouvoir profiter des services proposés par un data center avec une utilisation continue, il faudra prévoir une maintenance efficace allant du plus petit composant des machines physiques existant dans les salles informatiques jusqu'à la climatisation et l'aération.

Une bonne maintenance d'un data center donné permettra ainsi de garder les performances proposées par celui-ci. Ainsi le client sera satisfait des services qui lui seront offerts. Dans ce qui

suit, nous analyserons la consommation des data centers, des serveurs ainsi que des machines virtuelles.

2.1.1 Consommation des data centers

Alors que les ménages sont alertés sur les enjeux énergétiques et sensibilisés à la nécessité d'une meilleure gestion de leur consommation domestique en énergie, les entreprises elles trouvent du mal à adopter une politique efficace pour réduire leur propre consommation. Pour les data centers, certains préfèrent migrer vers une énergie verte qui s'avèrent rentable pour tenir un petit data center, mais le sera moins pour un moyen ou grand data center.

Adopter une énergie renouvelable pour tenir un data center en activité permanente revient à utiliser comme principale source énergétique l'énergie solaire, éolienne, hydromécanique et géothermique. Cette stratégie d'adoption d'une énergie renouvelable est certainement bénéfique, mais elle reste coûteuse et lente. Ce verrou technologique est lié au coût de stockage et au transport. D'autant plus qu'il est difficile d'utiliser ces sources, car, contrairement à l'énergie fossile ils ne sont pas toujours disponibles. Cela signifie que la demande d'énergie et de l'offre doit être adaptée, si nous voulons tirer pleinement parti de l'énergie verte. D'un autre côté, cette stratégie de l'énergie verte peut causer une augmentation de la latence d'accès aux services proposés et le coût de l'électricité. Afin de satisfaire les demandes du client, quelques configurations doivent être élaborées auparavant ce qui n'encourage pas forcément le fournisseur puisque cela engendrera d'autres coûts qui au final lui feront parfois perdre tous ses gains.[6]

Dans d'autres cas, un même fournisseur pourrait avoir plusieurs data centers. Varier les sources d'énergie mène dans la plupart des situations à une différence dans le taux de transmission d'informations d'un centre de données à un autre. Même si plusieurs techniques sont utilisées pour l'optimisation du trafic inter-data center, selon [7], il est préférable de les éviter au maximum, car peu importe l'optimisation à faire, éviter un trafic supplémentaire reste toujours le meilleur moyen pour limiter la consommation d'énergie.

2.1.2 Consommation des serveurs

Au sein d'un data center, nous pourrions trouver un nombre important de serveurs et que chacun de ces derniers héberge un nombre de machines virtuelles proportionnel à sa capacité. La

consommation d'énergie dans un data center donné est relative au besoin en énergie des serveurs pour maintenir une bonne performance des services hébergés.

Par conséquent, et dans le but de diminuer la consommation d'énergie d'un data center, nous devons optimiser l'utilisation des serveurs. Pour cela, diverses méthodes sont utilisées. Une méthode d'optimisation de l'utilisation des serveurs doit être élaborée afin que les machines virtuelles qu'ils hébergent ne nécessitent pas un trafic inter-serveur ou encore un trafic inter-data center.

Il faudrait aussi penser à répartir la charge de travail des serveurs comme proposé par [8], de sorte qu'il n'y ait pas, dans un même data center, un serveur sur lequel repose une charge de travail importante. Cela pourrait entraîner une augmentation de la température et donc un besoin d'une régulation à travers la climatisation.

Afin de limiter la consommation en énergie d'un centre de données, il est impératif de réguler le besoin en ressources de ses serveurs. Maintes simulations ont été faites pour connaître le besoin et la consommation d'énergie annuelle de refroidissement des centres de données. Une analyse de son impact réalisée par [9] a été élaborée afin de calculer la température idéale pour le maintien du bon fonctionnement des serveurs.

La qualité de services des centres de données est un critère important pour le client afin de faire le choix du fournisseur qui possède le data center qui convient le mieux à son utilisation. L'accord de niveau de service entre le fournisseur et le client restent le seul moyen pour garantir une continuité des services proposés. Le fournisseur devrait donc, essayer de réduire la consommation d'énergie de ses data centers tout en gardant ses serveurs en parfaite harmonie avec ce que le consommateur de service demande.

Si un serveur atteint sa capacité maximale en termes de mémoire, il pourrait ralentir le trafic des données. Par conséquent, une migration de la machine virtuelle s'impose vers un autre serveur qui pourrait maintenir la performance souhaitée sans pour autant pénaliser les services du nouveau serveur. Selon [10], cette migration atténuera la montée en charge de travail du serveur et doit absolument être automatique, rapide et invisible pour le client. Selon [11], si aucun des serveurs de ce data center ne pourrait héberger cette machine virtuelle, nous pourrions envisager de procéder à la migration vers un autre data center. Cette manipulation sera certes plus coûteuse en termes d'énergie, mais elle sera certainement bénéfique à long terme.

La gestion interne d'un data center est donc assez importante pour limiter la consommation d'énergie de ce dernier. Sa gestion démarre principalement par la gestion des ressources internes d'un serveur. Par conséquent, il faudra contrôler la consommation des machines virtuelles hébergées par ces serveurs.

2.1.3 Consommation des machines virtuelles

Les machines virtuelles sont ceux que le consommateur de service a comme repère pour la qualité de service d'un data center. De ce fait, le fournisseur devrait tout miser sur ces VMs afin de satisfaire au maximum les attentes de son client.

Étant donné que les machines virtuelles sont ceux qui définissent les besoins en énergie d'un serveur et donc d'un data center, ils existent plusieurs méthodes pour prévoir la puissance énergétique nécessaire pour le maintien de la globalité des machines virtuelles et ainsi partager les machines virtuelles de façon à minimiser la consommation d'énergie. Sauf que cette même action de prévision pourrait s'avérer coûteuse et fait fuir les fournisseurs puisque les gains ne sont pas considérables.

Dans une étude élaborée dans [12], les chercheurs ont modélisé une fonction pour le calcul de la consommation d'énergie. Cette modélisation s'avère être une tâche nécessaire lors de la configuration ou de l'assignation d'une machine virtuelle à un serveur donné, la mise en marche d'une nouvelle machine ou d'une migration de celle-ci. La gestion des opérations reliées aux machines devrait donc être bien étudiée de telle sorte qu'aucun imprévu ne pourrait bloquer l'utilisation des services à travers la VM.

Beaucoup d'efforts a été fait dans le but de minimiser la consommation d'énergie des data centers. Cependant, la plupart des approches manquent l'analyse des applications qui roulent au niveau des machines virtuelles. La variation de charge entre les différentes machines virtuelles d'un même data center voire d'un même serveur ne sont pas étudiées. Certes, la migration directe pourrait aboutir à de bons résultats, néanmoins, cette technique est elle-même consommatrice d'énergie et pourrait être évitées s'il y'aurait auparavant une bonne étude prévisionnelle. Pour remédier à ces faiblesses, une approche élaborée dans [13] a proposé une nouvelle architecture pour l'allocation des ressources de cloud . Ainsi nous saurons pour chaque machine virtuelle quelle est le seuil qu'elle pourrait atteindre en termes de demande de ressources pour ainsi fixer un seuil de de

consommation d'énergie et si cette VM sera colocaliser avec d'autres machines virtuelles sur un même serveur apte à tenir une performance stable pour tout ce qu'il héberge.

2.2 Réduction de l'empreinte carbone

L'empreinte carbone d'un data center est la quantité de gaz à effet de serre, principalement le dioxyde de carbone, rejeté après avoir effectué des activités dans le but de présenter ses services. La mesure de l'empreinte carbone, selon [14], est défini par la quantité d'énergie consommée pour alimenter le data center principalement la puissance des équipements informatiques, de la climatisation et de l'éclairage.

L'empreinte carbone dépend directement de la nature de la source d'énergie. En effet, utiliser comme source d'énergie une énergie renouvelable baissera considérablement l'empreinte carbone de ce data center. De plus, l'endroit où se situe ce data center est important, car la localisation de ce dernier joue un rôle important pour définir la température de maintien. Le besoin en climatisation diminue dans un environnement froid et inversement.

Réduire la consommation d'énergie d'un data center pourrait parfois aboutir à une dégradation des performances et de la qualité des services. En effet, certaines techniques élaborées telles que la migration à chaud pourront avoir, selon [15], un impact négatif sur la dégradation des performances. De plus lors de la pratique de ces techniques dans le but de réduire l'énergie consommée, c'est tout le contraire qui se passe. En effet, utiliser l'approche adéquate au moment convenu est obligatoire pour réduire la consommation d'énergie.

Au tout début de l'approche à adopter permettant la réduction de la consommation d'énergie, il est nécessaire de trouver une bonne configuration initiale pour le placement des machines virtuelles au sein de chaque serveur.

L'emplacement d'un data center pour un fournisseur est un choix qui se fait selon plusieurs critères. L'un des critères les plus importants est le budget. Certes un data center dans un climat frais pourrait participer à la réduction de la consommation d'énergie, néanmoins, cela pourrait coûter plus cher pour certains, à cause de la maintenance, mais aussi en vue de la distance qui sépare le data center des destinations utilisant les services cloud.

En ce qui concerne les serveurs, leurs localisations se fait en fonction de la taille des data centers. En effet plus le data center est grand et plus nous aurons de la place pour mettre les serveurs. Il

faudra bien sûr ne pas l'encombrer, car plus nous aurons des serveurs et plus il y'aura du trafic réseau dans le data center et l'énergie consommée pour la climatisation sera par conséquent plus élevée.

Une fois que toute l'architecture de tous les data centers sera mis en place, nous pourrions nous consacrer au placement des machines virtuelles. Cette étape est indispensable pour la définition de l'empreinte carbone d'un data center donné. En effet, la charge de travail de ce dernier dépend du trafic élaboré entre le serveur contenant la VM et le client. Il est impératif que la qualité du service proposé respecte les accords de niveau de service. Plus la charge de travail des machines virtuelles augmente et plus il aura besoin de ressources. Par conséquent, lui acheminer les ressources nécessaires serait primordial. Mais le problème consiste dans le fait que le serveur qui héberge cette VM-là pourrait ne pas être capable de satisfaire les besoins en ressources, à cause d'une autre machine qui les monopolise. C'est pour cela qu'une configuration initiale, qui consiste au bon placement des machines virtuelles dans l'environnement infonuagique, s'impose.

2.3 Placement des machines virtuelles

Trouver la bonne approche pour le placement des machines virtuelles au sein d'un data center est une étape importante pour minimiser l'empreinte carbone. Plusieurs études ont été élaborées dans ce contexte. Nous allons les présenter dans ce qui suit.

Les machines virtuelles sont généralement placées en fonction des exigences des locataires ou clients. Ces exigences, surtout au niveau de la qualité de services, rendent nécessaire un bon placement des machines virtuelles pour la réduction des coûts du réseau des centres de données. Le placement de ces VMs dépend de l'architecture globale du data center ainsi que du nombre de machines virtuelles à placer. S'il existe plusieurs VMs dans un même centre de données, nous aurons bien évidemment un trafic réseau plus élevé.

De ce fait, trouver un bon placement pourrait réduire les coûts du data center en énergie. Dans [16], les chercheurs ont défini le coût général d'un centre de données comme étant la somme du coût du trafic réseau (N-cost) ainsi que le coût de machines physiques (PM-cost) principalement composées des serveurs. En prouvant que le placement des machines virtuelles au sein d'un data center est un problème NP-difficile, l'approche élaborée dans cette étude, a divisé le problème en trois phases. Tout d'abord, la minimisation du coût du trafic réseau. Cette opération est elle-même prouvée

comme étant un problème NP-difficile. Puis, commencer par fixer la valeur de N-cost afin de minimiser PM-cost. Enfin, la troisième phase consiste à minimiser les deux variables, à savoir, le coût du trafic réseau ainsi que le coût des machines physiques. Pour cela, cette étude procède par ordre de priorité. Les demandes avec un nombre de machines virtuelles plus élevé sont prioritaires, puisque dans le cas où ces derniers seront partagés, le fournisseur du service a besoin de plus de machines physiques pour les héberger et le trafic réseau sera plus élevé. Le compromis entre N-cost et PM-cost a été assuré par un algorithme heuristique basé sur la recherche binaire.

Dans un contexte où les applications sont complexes nécessitant plusieurs machines physiques, les machines virtuelles doivent être réparties d'une manière à ce que chaque serveur tourne d'une manière ordinaire et les pics en utilisation seront gérés. Dans [17], les chercheurs ont travaillé sur ce dernier point, et la performance des services offerts était la priorité. L'approche élaborée a mis en évidence la montée en charge du travail d'une machine virtuelle et le devoir du fournisseur de répondre à ses besoins. En effet, une VM colocalisée avec une autre VM d'un autre client dans un seul serveur pourrait causer certains soucis au niveau des performances des services proposés. Cette contrainte a donc été considérée pour être certain qu'une colocalisation de ces deux VMs n'aura aucun impact sur la performance de ces derniers. Le but est donc de maximiser la fiabilité ainsi que la disponibilité des services tout en minimisant la violation de l'accord du niveau de service. Le principe part du fait que l'utilisation des ressources doit être équilibrée. L'algorithme utilisé démarre donc d'une liste de machines virtuelles ordonnées et essaye de placer les VMs une par une parmi un ensemble fixe de machines physiques. Le placement se fait par rapport au serveur approprié et donc celui qui offre les ressources nécessaires à chaque VM.

Avec l'émergence du cloud, les applications doivent bénéficier de l'élasticité au niveau des ressources. Garantir cette élasticité est un défi pour le fournisseur. La qualité de service est primordiale pour une machine virtuelle qui demande des ressources variées au cours du temps. À cette fin, une étude élaborée par [18] a présenté une approche, qui se base sur des données dynamiques des machines virtuelles. Les décisions de placement suivent trois méthodes probabilistes pour l'évaluation de l'aptitude de l'hôte et ses demandes de ressources. Une estimation est faite lors du placement des machines virtuelles. Cette estimation vise à reconnaître la demande en ressources des VMs en fonction de l'utilisation ancienne et actuelle de ces derniers ou des applications du même type que celles hébergées par ces VMs. Pour éviter une sous-estimation ou une surestimation, l'approche élaborée utilise une méthode pour la réservation des ressources. Dans

ce cas, si au cours de son utilisation, l'application aura besoin de plus de ressources, elle utilisera les quantités de ressources souhaitées dans la limite de celles réservées. Une fois qu'elle aura fini son utilisation et que celle-ci revient à son ordinaire, toutes les ressources allouées redeviendront réservées.

Le bon placement des machines virtuelles sur les machines physiques participe à la réduction de la consommation d'énergie d'un data center. Mais l'action en elle-même est consommatrice d'énergie. Certaines études ont travaillé sur la réduction des coûts de placement. Une étude réalisée dans [19] a présenté un système de placement des VMs basé sur la théorie des graphes. L'algorithme prend en considération la demande du trafic inter-VM, ainsi que le trafic inter-serveurs pour les applications partagées entre plusieurs serveurs. Ainsi, à travers cette méthode, il est possible de déterminer le nombre de serveurs nécessaires pour héberger toutes les VMs et pouvoir bénéficier de leurs services sans pour autant diminuer leurs performances.

Plusieurs autres études ont accordé la priorité de minimisation des connexions entre les machines virtuelles lors du placement des VMs. L'étude élaborée dans [20], présente un algorithme qui consiste à trouver une combinaison de placement des machines virtuelles capables de minimiser le trafic réseau inter-VM. Dans cette approche, le modèle pour la gestion des machines virtuelles peut être centralisé ou bien distribué. Dans le cas où il est centralisé, l'algorithme commence par placer la VM jouant le rôle du nœud central de sorte que les autres serveurs, s'ils ont à communiquer avec le serveur central, aient le même coût en termes de trafic réseau. Ensuite, pour le placement des autres VMs, le but consiste à ce que le trafic avec le nœud central soit minimal et pratiquement le même pour toutes les autres machines virtuelles tout en priorisant ceux qui ont une communication plus fréquente. Pour le cas où le modèle est distribué, un algorithme sera appliqué, similaire à celui utilisé pour le modèle centralisé, mais qu'à chaque fois qu'on place une machine virtuelle, cette dernière sera considérée à son tour comme le nœud central et ainsi de suite jusqu'à la fin du placement de toutes les VMs.

Les algorithmes de placements de machines virtuelles ne cessent d'évoluer afin de s'adapter aux nouveaux problèmes pour la gestion des ressources. Dans [21], une pratique dynamique a été développée permettant d'adopter une configuration initiale de placement des VMs, avec comme objectif principal, la minimisation du coût des migrations à chaud des machines virtuelles d'un serveur vers un autre. L'approche consiste tout d'abord à trouver une solution réalisable où toutes

les contraintes et les exigences du client sont satisfaites. Ensuite une heuristique sera appliquée pour essayer de trouver la combinaison qui ne viole aucune contrainte. La priorité de placement des VMs dans les data centers consiste à l'utilisation minimale de machines physiques. C'est donc le nombre de serveurs utilisés qui détermine la consommation d'énergie d'un data center. Le trafic inter-VM, pour cette solution, n'a pas été pris en considération étant donné qu'il a été supposé que son coût soit négligeable devant le coût de maintien d'une machine physique en marche.

Le problème de placement des machines virtuelles pour la réduction de la consommation d'énergie et l'empreinte carbone se résume, pour certains, à un problème d'un surplus d'utilisation de machines physiques dans un data center. Mettre en veille ces serveurs physiques est devenu le principal objectif. Selon [22], faire le bon choix du serveur à mettre en veille a une grande importance. Pour cela, l'approche suit une topologie similaire à la topologie de réseau en arbre. L'algorithme parcourt ainsi l'arbre et tente de mettre au niveau du sous-arbre de chaque nœud parcouru une machine virtuelle. Une estimation du nombre de VMs pouvant être accepté par un même serveur sera alors faite pour chaque nœud. S'il existe assez de ressources et de bande passante, le nœud sera validé pour ainsi allouer cette machine virtuelle au serveur actuel. Pour cette approche, plus la capacité des serveurs est grande et plus la méthode est efficace.

D'autres études se sont basées sur le fait que les machines virtuelles les plus actives doivent rester au niveau de la même machine physique. Ainsi la notion de répartitions de charge de travail au niveau des serveurs n'est pas considérée. L'étude faite dans [23] a adopté cette approche. En effet, éliminer le fait que toutes les machines physiques travaillent tout le temps avec une bande passante assez élevée serait la cause de l'augmentation de la consommation d'énergie. D'un autre côté, il est nécessaire de donner la priorité aux machines virtuelles collaborant ensemble appelées machines virtuelles complémentaires. Ces VMs complémentaires réunies sur une même machine physique entraînant la communication inter-VMs profiteront d'une bande passante élevée et réduiront le trafic intra-data centers. Considérant la propriété aléatoire des besoins en ressources des machines virtuelles, cette approche calcule la probabilité de débordement des ressources par rapport à celle disponible pour ainsi satisfaire les besoins de tous les VMs

Le fait que les machines virtuelles ont un impact direct sur la consommation énergétique de la machine physique sur laquelle elles sont hébergées, dans [24], l'étude a accordé une importance quant à la réduction du besoin en énergie d'un data center à travers une bonne gestion des ressources

allouées aux machines virtuelles. En effet, le modèle proposé se base sur les exigences des clients lors de leurs requêtes pour l'allocation des nouvelles VMs. Chaque requête comporte des caractéristiques propres à elle. Si le client désire avoir plus ressources, le seul moyen pour le signaler serait de faire une requête. De ce fait, le client saura gérer lui-même les montées en charge de travail de son application et les fournisseurs n'auront plus à se soucier des coûts de l'approvisionnement des ressources. Après l'assignation des VMs aux serveurs, le système attend toute requête des clients. Si une machine physique ne pourra pas répondre aux besoins en termes de ressources, un algorithme élaboré dans [25], s'occupera de faire une migration à chaud de la machine virtuelle vers un autre serveur.

La gestion des ressources allouées pour les machines virtuelles est un point important pour la minimisation de la consommation d'énergie. Ainsi, lors du placement des machines virtuelles sur les machines physiques des data centers, il ne faut pas tenir compte des contraintes de chaque VM à part, mais plutôt des contraintes multiples d'un nombre important de VMs. Ainsi, trouver une combinaison pour le placement des machines virtuelles serait mieux que de placer les VMs une par une. L'approche élaborée dans [26], cherche à trouver une optimisation combinatoire afin de minimiser le nombre de machines physiques actifs. Afin de réaliser cette approche, deux phases seront nécessaires. La première phase est le placement statique des machines virtuelles. Pendant cette phase, toutes les VMs n'ayant pas une dépendance entre elles seront placées sur une même machine physique, mais différente de celle qui a une dépendance. Les serveurs hébergeant des VMs devraient être le plus loin possible les uns des autres, car plus la distance est grande et moins le trafic sera important. Celles qui auront des dépendances seront placées sur la même machine physique. Dans le cas où la machine physique a atteint son maximum en termes de ressources, une allocation d'un nouvel espace sera faite dans un autre serveur pour la nouvelle machine virtuelle. Une fois que toutes les VMs ont été bien placées, la deuxième phase entre en jeu. Cette phase n'est autre que le placement dynamique des machines virtuelles. Chaque nouvelle VM à placer dépendra du trafic réseau au sein du data center. Si nous prévoyons un trafic élevé pour cette VM, nous devons la placer sur la machine physique où le trafic est important. Un tel choix est dû au fait qu'avec les machines virtuelles à trafic élevé, nous aurons la certitude qu'elles seront utilisées quotidiennement et que les risques pour qu'elles sous-utilisent les ressources allouées sont moindres. De plus, les migrations inter-serveurs seront minimisées.

Maintenant que les data centers sont en train d'offrir de plus en plus de services, il est devenu nécessaire pour certaines industries de se diriger vers l'externalisation de leurs données vers divers centres de données. Certaines informations, considérées comme confidentielles, resteront au niveau local. Cette séparation engendrera forcément un trafic plus élevé que si elles étaient regroupées dans un même data center. Le fournisseur devra donc faire en sorte de minimiser au maximum le trafic inter-data center. Dans [27], une étude a détaillé le problème des données partagées sur plusieurs machines virtuelles dans un même data center ou entre plusieurs data centers. Le placement des VMs dans ce cas-là priorise la minimisation du trafic inter-VM ainsi que la bande passante dans le réseau du data center. Considéré comme un problème de Bin-Packing dans cette approche, l'approche utilisée considère qu'une demande de réapprovisionnement en ressource n'est pas une étape systématique. En effet, chaque requête doit être bien étudiée avant d'être acceptée, car cela pourrait obliger à faire une migration de cette machine virtuelle vers un autre serveur et donc une consommation d'énergie non prévue. Le choix de la machine physique qui va héberger la machine virtuelle sera fait en fonction de la consommation du processeur du serveur. Un bon choix est caractérisé par la non-violation de l'accord du niveau de service. Pour l'étape de la migration d'une VM, le serveur choisi est celui qui aura le minimum d'énergie consommée.

D'autres études considèrent qu'un bon placement de machines virtuelles repose sur le temps de réponse rapide de ces dernières. Si le temps de réponse est court alors le trafic sera moins dense et donc l'énergie consommée sera moins importante et l'impact sur l'environnement sera minimal. L'algorithme présenté dans [28] cherche à trouver un ajustement permettant à chaque application d'avoir un temps de réponse idéal offert au niveau de la machine physique. Si le client désire améliorer le temps de réponse, un algorithme glouton entre en jeu et cherche à déterminer la nouvelle machine physique adéquate. Si cet algorithme donne une solution non faisable, un recours au premier algorithme d'ajustement sera nécessaire comme si c'est une nouvelle machine virtuelle à placer.

Enfin, une dernière étude élaborée dans [29], a proposé une approche pour le placement des machines virtuelles basée sur un modèle qui prend en considération les principaux contributeurs de la facture totale d'électricité d'un data center, à savoir la puissance des ventilateurs, des serveurs ainsi que le système de refroidissement. Dans cette étude, l'objectif principal étant de diminuer l'empreinte carbone d'un data center, le problème de placement des machines virtuelles doit être relié directement à l'évaluation de la meilleure température pour le maintien des machines

physiques. Cette étude a été améliorée dans [30] par la modification du modèle pour la prise en considération du trafic réseau. Pour cela, un algorithme hybride mettant en jeu la recherche locale itérée avec l'algorithme de recherche tabou a été conçu. De ce fait, nous bénéficierons des avantages des deux algorithmes à savoir le fait d'élargir l'espace de travail offert par ILS, mais aussi le fait de rechercher efficacement la meilleure solution par l'algorithme de recherche tabou.

Cette étude a élaboré un modèle mathématique non linéaire. Ce modèle ayant pour objectif la prise en compte de tous les composants d'un data center qui définit le meilleur placement des machines virtuelles dans un environnement infonuagique. Le premier facteur est le facteur d'émission d'un data center qui forme le premier critère de choix pour le placement des VMs. En effet, chaque data center a une certaine consommation et besoin en énergie en fonction de son emplacement. La considération de ce facteur est primordiale même si dans certains cas, il est impossible de prédire le comportement de ce data center. Il faut aussi savoir que parfois les data centers qui consomment le moins, ne sont pas réellement un bon choix pour héberger les VMs. Il est donc nécessaire de tester toutes les configurations possibles.

Dans ce même modèle, la planification des machines virtuelles prend en compte la puissance des équipements informatiques. En effet, chaque équipement participe de près à la définition de l'empreinte écologique d'un data center, allant des machines virtuelles jusqu'aux commutateurs qui relient les différents châssis ainsi que les systèmes de ventilation. Les VMs étant les seules variables parmi tous les équipements, ce sont ces dernières qui définissent la puissance nécessaire pour leurs alimentations. La température dans un environnement infonuagique caractérise la qualité des services offerts par les machines virtuelles. En effet, si la température dépasse le seuil, un problème pourrait se manifester jusqu'à détérioration des services. Afin de garder la température inférieure au seuil, le fournisseur de service élaborera un système de refroidissement qui réagira en fonction des changements. Dans le modèle cité précédemment, nous chercherons la valeur du seuil de température qui maintiendra le niveau de service. De plus, il tentera de minimiser le trafic réseau intra-data centers élaboré entre les machines virtuelles.

La figure 2.1 illustre l'architecture d'un data center utilisée pour la modélisation du problème. Elle démarre à partir d'un commutateur principal. Des commutateurs d'agrégation reprennent la relève afin d'assurer le trafic réseau entre les différents composants. Les châssis, eux, sont ceux qui regroupent les serveurs.

Cette architecture se reproduit pour tous les data centers. Une fois que l'environnement a été mis en place, l'algorithme de placement des machines virtuelles dans les data centers sera prêt à démarrer. Cette méthode procèdera d'une manière statique, étant donné qu'il ne considère pas le changement des ressources physiques des VMs.

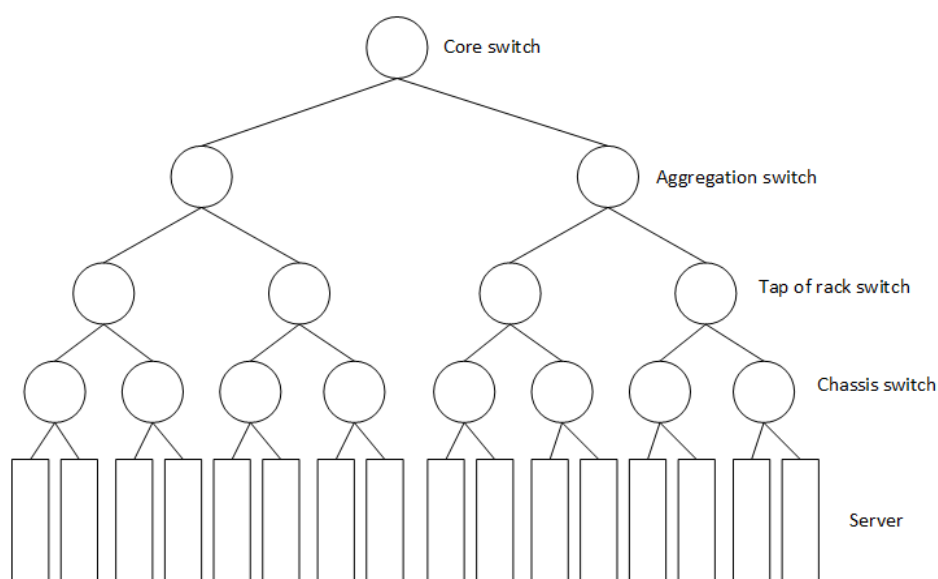


Figure 2.1 Topologie d'un data center [30]

Dans cet environnement infonuagique, un système de gestion des machines virtuelles sera responsable pour l'affectation de chaque VM dans le data center adéquat et en fonction du client qui a présenté sa demande. La figure 2.2 présente un aperçu de cet environnement où le VMMS s'occupera de la gestion des machines virtuelles.

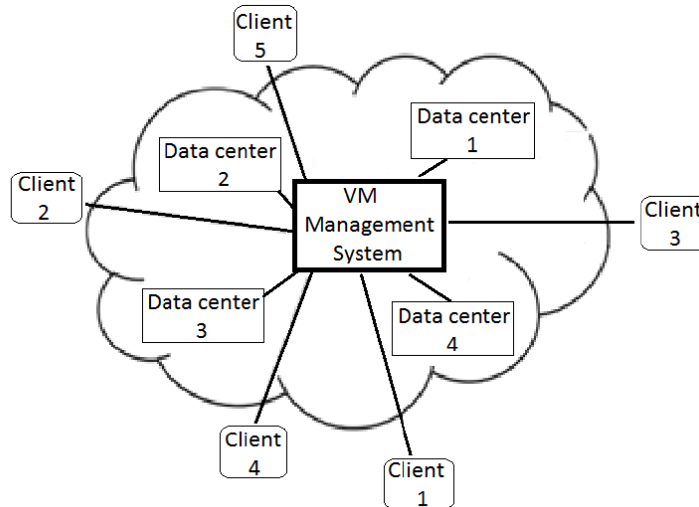


Figure 2.2 Environnement infonuagique [29]

L'algorithme responsable du placement des VMs dans cet environnement sera lancé dans le VMMS, puisque ce dernier a une vision plus globale de l'ensemble des composants des data centers.

2.4 Heuristiques utilisées pour le placement des machines virtuelles

En optimisation combinatoire, une métaheuristique est une méthode approchée qui permet de trouver, pour un problème NP-difficile, une solution réalisable pas forcément optimale en un temps polynomial. En ce qui concerne le problème de la planification des machines virtuelles dans un data center, trouver la meilleure combinaison de placement prendrait un temps énorme. De plus, en vue du dynamisme dans l'utilisation des applications, nous serons obligés, à chaque fois qu'une exigence s'ajoute, de refaire tout le travail. Trouver une solution rapide devient donc le seul moyen pour planifier un bon placement des machines virtuelles pour ainsi faire profiter des services proposés et minimiser l'énergie consommée ainsi que l'empreinte carbone.

Nous pourrions classer les métaheuristiques en trois catégories à savoir la recherche locale, les métaheuristiques à population ainsi que les méthodes hybrides. La figure 2.3 présente ces trois catégories.

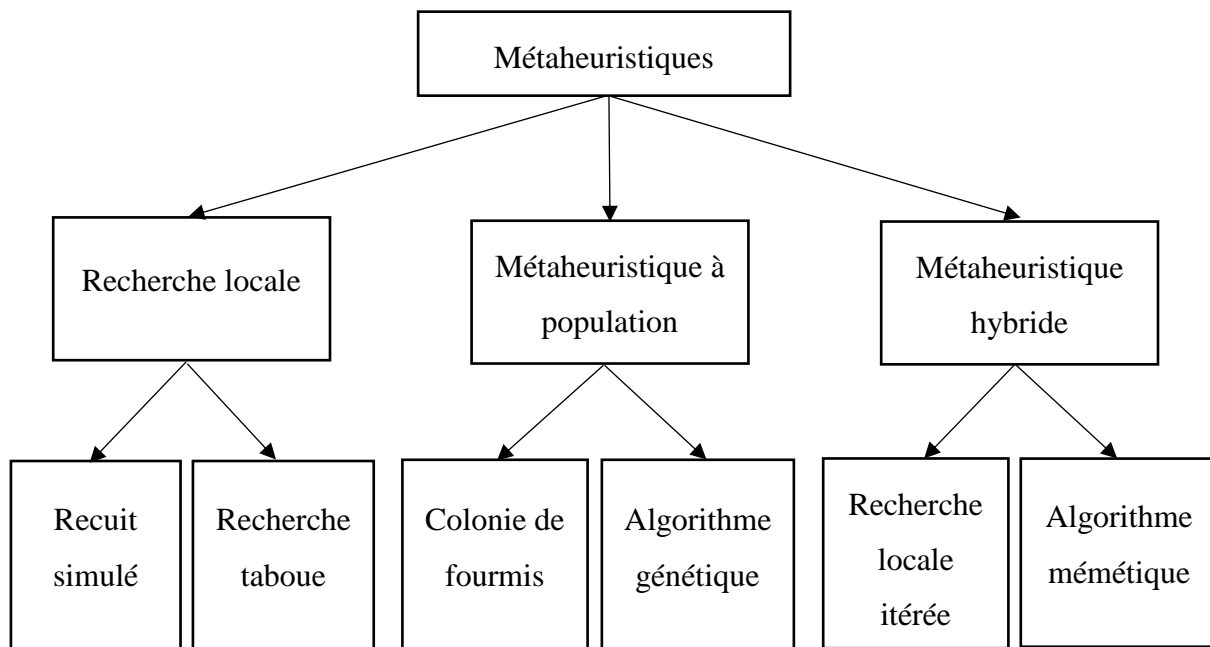


Figure 2.3 Classification des métaheuristiques

Plusieurs algorithmes de métaheuristique ont été élaborés afin de résoudre le problème de placement des machines virtuelles. Dans ce qui suit, nous détaillerons l'algorithme génétique ainsi que l'algorithme mémétique avec l'algorithme de recherche tabou et enfin le recuit simulé. Ces algorithmes seront utilisés dans ce travail pour la résolution du problème de placement des machines virtuelles dans un environnement infonuagique.

2.4.1 Algorithme génétique

L'algorithme génétique est un algorithme évolutionniste basé sur des phénomènes biologiques. En effet, en commençant par une solution initiale, nous appliquons certaines opérations tirées de la génétique telles que la sélection, le croisement et la mutation afin de générer de nouvelles générations qui représentent l'ensemble de solutions. Chaque solution candidate possède un nombre de propriétés sur lesquelles nous appliquerons les opérations citées. L'apparition de nouvelles générations signifie que de nouvelles solutions ont été créées.

C'est lors de la sélection qu'on gardera les solutions qui, selon des critères fixés préalablement, seront les plus adaptées à faire apparaître de meilleures solutions.

Le croisement dans l'algorithme génétique, représenté dans la figure 2.4, est le fait d'obtenir à partir de deux chromosomes, deux nouveaux chromosomes. Lors de cette étape, un ou plusieurs points de croisement seront fixés et qui serviront par la suite d'extrémités pour l'échange de gènes entre les deux chromosomes.

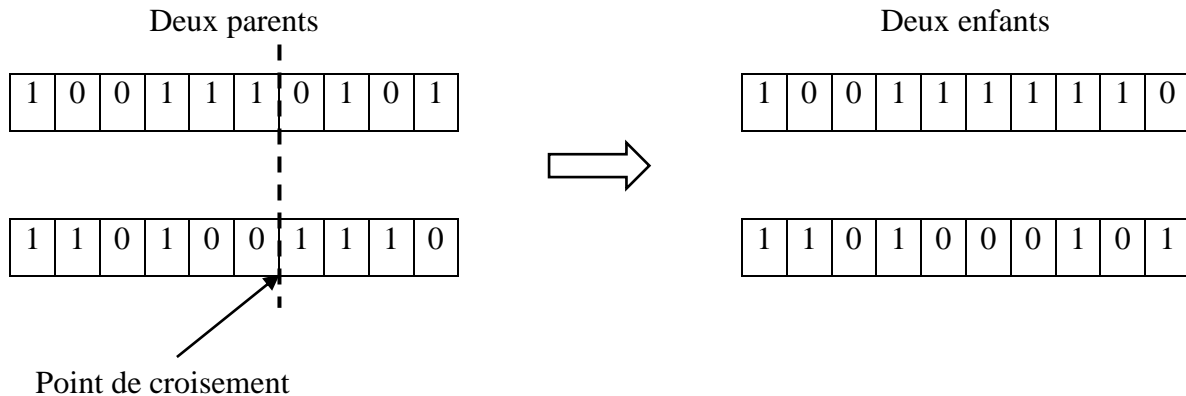


Figure 2.4 Représentation du croisement dans l'algorithme génétique

Dans le but d'éviter une convergence rapide de l'algorithme génétique, nous utiliserons la mutation. La figure 2.5 illustre son principe qui consiste à modifier un gène pris aléatoirement d'un chromosome donné. Afin de bénéficier des résultats obtenus des opérations de croisement et de sélection, la mutation ne doit pas être appliquée sur tous les chromosomes de la génération obtenue.

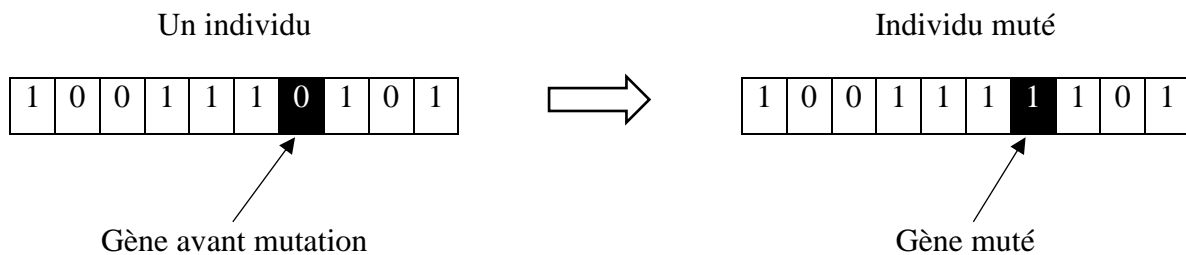


Figure 2.5 Représentation de la mutation dans l'algorithme génétique

Lors des opérations de croisement ou de mutation, il y a de fortes chances de perdre les chromosomes ayant données une bonne solution. Afin de conserver les meilleurs individus, nous utilisons lors de l'opération de sélection, l'élitisme. Celui-ci consiste à faire passer ces meilleurs individus dans la génération suivante sans effectuer de modifications.

Parfois les nouvelles solutions ne seront pas réalisables. Une dernière étape permettra l'évaluation des individus et de choisir, selon des critères de faisabilité, quel individu choisir pour l'ajout dans la nouvelle population.

Si le nombre d'itérations fixé au préalable a été atteint, nous arrêtons l'algorithme et choisirons la meilleure solution parmi celles qu'on aurait générées.

2.4.2 Algorithme mémétique

L'algorithme mémétique est un algorithme hybride combinant à la fois l'algorithme génétique et un algorithme de recherche locale. En effet, après avoir réalisé toutes les étapes principales de l'algorithme génétiques à savoir la sélection, le croisement et la mutation, un algorithme de recherche locale sera élaboré sur chaque individu de la nouvelle génération créée.

L'avantage des algorithmes mémétiques est de bénéficier du fait que les algorithmes génétiques permettent de découvrir de nouvelles solutions alors qu'avec les algorithmes de recherche locale nous éviterons de converger rapidement. Parmi les algorithmes de recherche locale utilisés dans les algorithmes mémétiques figure l'algorithme de recherche tabou.

L'algorithme de recherche tabou est un algorithme basé sur l'exploration du voisinage afin de découvrir de nouvelles solutions. Parfois les solutions explorées ne minimisent pas la fonction objectif dans le cas d'une minimisation, et inversement. Malgré le fait que ces dernières n'améliorent pas le coût de la solution, nous sommes tentés de les accepter afin de donner la chance à la découverte de nouvelles solutions, qui à leurs tours pourront aboutir à une amélioration de la solution recherchée. Ce mécanisme permet donc d'éviter de se retrouver piégé dans un minimum local.

Aussi et dans le but de ne pas choisir le même voisin sans pouvoir en explorer d'autres solutions, une liste taboue doit être utilisée. Cette liste, de principe FIFO (premier arrivé, premier sorti), a pour objectif de mémoriser les positions des derniers mouvements. Ainsi, ces mouvements ne pourront plus être élaborés jusqu'à ce qu'ils sortent de celle-ci.

La taille de liste reste un paramètre à définir en fonction de la taille du problème ainsi que des résultats obtenus avec des listes de tailles différentes.

2.4.3 Recuit Simulé

Le recuit simulé est un algorithme de recherche local qui s'appuie sur un processus de recuit physique. Chaque mouvement élaboré pour l'amélioration de la solution considérée pourrait être accepté selon une loi de probabilité. La probabilité d'acceptation suit la règle de Métropolis. Elle sera donc faible si elle n'améliore pas la solution.

La figure 2.6 illustre l'opération d'acceptation d'une configuration donnée pour une température T donnée.

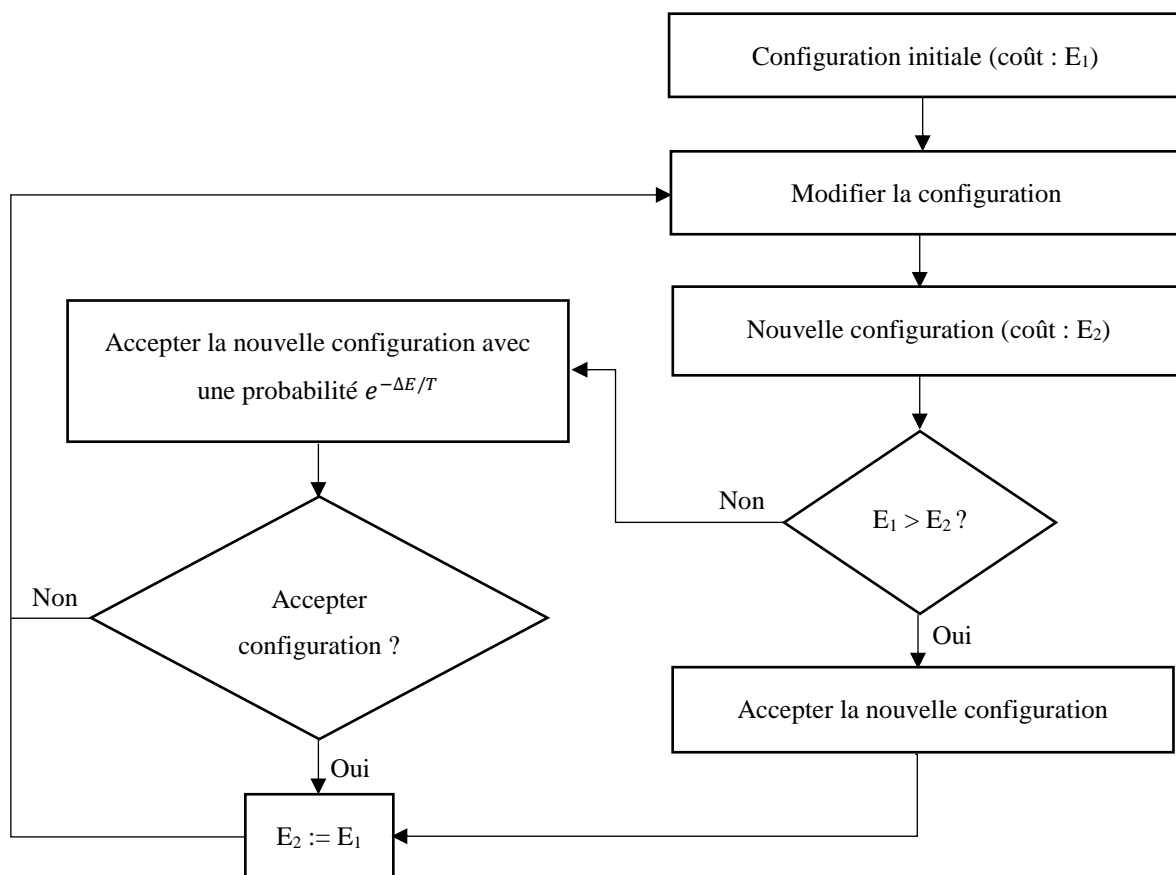


Figure 2.6 Acceptation d'une configuration dans le SA

Lors de l'élaboration du recuit simulé, plusieurs paramètres doivent être pris en compte. L'un des principaux facteurs est la température. Ce facteur dépendra du schéma de refroidissement qui sera calculé à partir des résultats de plusieurs solutions pris arbitrairement. La température diminuera en fonction du palier, mais aussi en fonction de l'amélioration du coût de la solution. Le taux de

décroissance de celle-ci sera fixé préalablement. Pour chaque palier, le nombre de configurations visitées est aussi un paramètre à définir au tout début de l'exécution de l'algorithme.

2.5 Synthèse des problèmes identifiés

Plusieurs études se sont intéressées au problème de planification des machines virtuelles dans un environnement infonuagique. Certaines d'entre elles ont considéré que le principal objectif serait de minimiser les dépenses encourues par le fournisseur de cloud [16]. Alors que d'autres accordent une réelle importance à l'amélioration des services offerts par leurs disponibilités ainsi que le respect de l'accord du niveau de service [17], [18]. Et enfin, quelques études se consacrent à la minimisation de l'énergie nécessaire pour l'alimentation de tout le data center. Cette minimisation concerne pour certains, la minimisation du trafic réseau par une bonne prédiction du nombre de serveurs capable d'héberger toutes les machines virtuelles [19]. D'autres considèrent cette minimisation par l'élaboration d'un système de gestion des requêtes provenant des clients dans le but d'allouer de nouvelles ressources [23, 28]. Enfin, une dernière considération de la minimisation de l'énergie consommée par un data center, présente les principaux facteurs participant à l'augmentation de la facture électrique pour minimiser le besoin énergétique tout en respectant l'accord de niveau de service [29]. Les intérêts pour ces sujets pourraient être combinés dans un même problème et donc optimiser l'empreinte carbone par la réduction de la consommation d'énergie tout en respectant l'accord de niveau de service entre le fournisseur et le client.

Dans le travail réalisé dans [29], Justafort et al. se sont intéressés à cette combinaison par la conception et l'implémentation d'un algorithme de métaheuristique pour le placement des machines virtuelles. L'optimisation se présente par un modèle complet permettant de réunir toutes les contraintes possibles reliées à la consommation de tous les composants du data center. L'algorithme utilisé est une hybridation entre la recherche locale itérée et l'algorithme de recherche tabou. De plus, l'impact écologique sur l'environnement a été traité par un modèle légèrement amélioré par rapport à celui élaboré dans [29], afin de prendre en considération le trafic réseau pour une meilleure utilisation des ressources des data centers [30].

La température étant un facteur essentiel pour la définition de la puissance requise pour le bon fonctionnement d'un data center, nous nous sommes intéressés à trouver la température idéale du point de vue consommation d'énergie ainsi que du maintien du niveau de service. En effet, les

systèmes de ventilation et de refroidissement sont considérés comme les principales sources d'augmentation de la facture d'électricité et de l'empreinte écologique. Le modèle implémenté vise à régulariser la température sans pour autant détériorer la qualité des services offerts.

Dans ce travail, nous allons concevoir et implémenter un algorithme mémétique avec comme fonction de coût le modèle élaboré par Justafort et al. dans [30]. Cet algorithme hybride représente une combinaison entre l'algorithme génétique et l'algorithme de recherche tabou.

L'évaluation des résultats de l'algorithme élaboré est faite par une comparaison des résultats obtenus avec la méthode exacte pour les instances de petite taille. Puis, pour les instances de grande taille, nous avons comparé avec les résultats obtenus par le recuit simulé qui lui sera implémenté dans ce travail ainsi qu'avec l'algorithme de recherche tabou itérée implémenté dans [30].

CHAPITRE 3 APPROCHE PROPOSÉE POUR LA RÉOLUTION DU PROBLÈME

Afin de résoudre le problème de placement des machines virtuelles dans un data center, plusieurs étapes doivent être réalisés. En effet, étant donné que la réduction de l'empreinte écologique est l'objectif principal pour le VMPP, nous utiliserons un modèle mathématique avec comme fonction de coût l'empreinte carbone qu'on cherchera à minimiser.

Comme cité dans le chapitre précédent, le modèle élaboré dans [29] sera le modèle à utiliser pour la résolution de ce problème étant donné qu'il regroupe tous les paramètres à prendre en considération pour le calcul de l'empreinte carbone.

Dans ce chapitre, nous définirons les principaux paramètres pour la modélisation du problème. Ensuite nous détaillerons la procédure qui mènera à la réduction de l'empreinte carbone. Enfin, nous présenterons les différents algorithmes de métaheuristique utilisés pour la résolution du problème de placement des machines virtuelles.

3.1 Modélisation du problème

Le modèle mathématique utilisé pour la résolution du problème de planification de machines virtuelles dans un data center est un modèle non linéaire incluant plusieurs contraintes ayant comme objectif principal la réduction de l'empreinte carbone.

Les machines virtuelles, dans ce contexte, sont les entités sur lesquelles repose tout le travail. En effet, chaque machine virtuelle a des caractéristiques propres à elle, essentiellement le processeur, la mémoire ainsi que stockage. Les serveurs de leur côté auront une capacité minimale pour l'hébergement de toutes les VMs. Ils auront pratiquement les mêmes caractéristiques que les machines virtuelles pour faciliter l'étape de placement des VMs. Ainsi lors de cette étape, la quantité de ressources disponible diminuera en fonction de la taille des machines virtuelles hébergées.

L'énergie consommée par un data center sera déduite à partir des châssis qui hébergeront les serveurs. Seuls les serveurs actifs participeront au calcul de l'empreinte carbone. À cela, sera additionné la consommation de la climatisation pour la régularisation de la température au sein du

data center ainsi que le trafic réseau intra-data centers créé par les machines virtuelles complémentaires. Bien entendu chaque data center a son propre facteur d'émission qu'on doit prendre en compte pour garantir le bon choix du data center qui héberge les VMs.

3.2 Réduction de l'empreinte carbone

Cette section sera consacrée pour la présentation des principaux facteurs qui participeront à la réduction de l'empreinte carbone. Ces facteurs sont un des intervenants dans le modèle mathématique utilisé.

Nous présenterons la manière qui permettra de déterminer la température idéale. Puis, nous détaillerons comment faire pour réduire le trafic réseau intra-data centers.

3.2.1 Détermination de la température idéale

Réguler la température au sein d'un data center est une étape importante lors de l'utilisation des machines virtuelles. En effet, lors de l'augmentation de la consommation des services ou l'atteinte d'un pic d'utilisation par le client, il faudra gérer les ressources allouées afin de garder un bon niveau de service. Pendant que les machines virtuelles seront utilisées, les serveurs ainsi que les châssis auront tendance à chauffer. Afin de garder un bon niveau de service tout au long de l'utilisation, il est préférable de trouver une température seuil.

La température varie d'un châssis à un autre. En effet, elle varie en fonction de la taille des châssis, du nombre de serveurs hébergés, ainsi que du nombre de ressources utilisées. Ainsi, lors de l'étape de placement, il faudra penser à laisser une certaine marge par rapport aux ressources utilisées. En effet, si les serveurs atteignent leurs saturations et donc leurs limites de ressources alors la température interne augmente et donc la température des châssis suivra par la suite.

La température initiale étant fixée à 15° Celsius et la température maximale tolérée de 35° Celsius [30], nous tenterons de trouver la température idéale dans un intervalle [15° Celsius – 35°Celsius]. Certes 15° Celsius est la température parfaite pour assurer un bon niveau de service, néanmoins, le fait d'augmenter la température permettra de réduire l'électricité globale consommée par un data center mais aussi d'améliorer l'indicateur d'efficacité énergétique de celui-ci sans pour autant détériorer les services proposés.

Ainsi, pour augmenter la température d'un data center il suffirait de réduire la puissance de ventilation sans dégrader les performances des services offerts. Dépasser le seuil de la température pourrait engendrer une détérioration de la qualité de service puisqu'une relation directe existe entre l'augmentation de la température et la QoS. Pour cela, à chaque fois que nous plaçons une VM dans un serveur, nous essayerons d'utiliser une puissance minimale de ventilation nécessaire pour le bon fonctionnement de la machine virtuelle. Nous tenterons ainsi de trouver le serveur qui minimisera au maximum l'énergie consommée pour le refroidissement. Cette étape se reproduira jusqu'à ce que le serveur atteigne le seuil maximal toléré pour éviter toute saturation.

3.2.2 Réduction du trafic réseau

Le trafic réseau dans un data center est toute communication élaborée avec les composants de ce dernier. Généralement, les informations circulent sous forme de requêtes de la part du client pour la récupération des informations et dans le sens contraire pour répondre à ces requêtes. Par ailleurs, une liaison pourrait se créer entre les machines virtuelles d'un même data center. Cette liaison engendre un trafic réseau intra-data centers.

Le trafic réseau intra-data centers devient plus important si les machines virtuelles complémentaires sont dans deux serveurs différents et plus si elles seront dans deux châssis différents. Le cas où les machines virtuelles communiquent entre elles en étant dans deux data centers différents n'est pas considéré vu que cela engendra des coûts encore plus élevés.

Une communication entre deux machines virtuelles n'appartenant pas au même serveur ni au même châssis devrait être le dernier recours du placement des VMs complémentaires. Si nous aurons à séparer deux machines virtuelles complémentaires sur deux châssis différent, l'information pourrait passer par le commutateur principal du data center et ainsi augmenter le trafic réseau. Dans ce travail, nous essayerons de minimiser ce genre de trafic dans le but de minimiser la consommation énergétique.

Il faut donc maximiser la colocalisation des machines virtuelles complémentaires au niveau du même serveur. Si cela ne le permet pas pour des causes de ressources indisponibles ou de saturation du serveur, il faudrait passer au prochain niveau de communication jusqu'à être obligé à reconsidérer le placement initial.

L'objectif est donc de restreindre le trafic réseau inter-VMs seulement au niveau des serveurs ou au niveau du châssis dans le cas où un serveur tout seul ne pourrait pas héberger les machines virtuelles en question.

3.3 Résolution du problème

Afin de résoudre le problème de planification des machines virtuelles dans un data center, un algorithme de métaheuristique doivent être mis en place. En effet, le principal algorithme à utiliser dans ce travail est l'algorithme mémétique qui est un algorithme hybride. Cet algorithme étant une combinaison entre l'algorithme génétique et une recherche locale, nous élaborerons en premier lieu une solution avec l'algorithme génétique. Ensuite, ce dernier sera combiné avec la recherche tabou. Enfin, nous essayerons de résoudre le VMPP en utilisant le recuit simulé, et ce dans le but de comparer par la suite les résultats obtenus.

Cette section sera consacrée à la présentation de la méthode de génération de la solution initiale. Ensuite nous présenterons l'algorithme génétique suivi de l'algorithme mémétique et le recuit simulé.

3.3.1 Génération de la population initiale

Les fichiers de données en entrées sont au nombre de deux. Le premier fichier donne des informations sur l'environnement composé d'un ou plusieurs data centers ainsi que tous les composants qu'ils hébergent. Le deuxième fichier inclut les informations sur les machines virtuelles ainsi que les liaisons entre elles.

À partir de ces informations, nous créerons un nombre de solutions initiales que nous essayerons de varier le plus possible grâce aux fonctions aléatoires. La priorité, lors de ce placement, est essentiellement que les machines virtuelles complémentaires minimisent au maximum le trafic réseau tout en restant dans le même data center. La deuxième sera de maximiser les serveurs passifs c'est-à-dire les serveurs qui n'hébergent aucune machine virtuelle. Bien évidemment, le respect de la disponibilité des ressources des serveurs doit être respecté. Enfin, la dernière priorité concerne les data centers dans le cas où il existe plus d'un dans l'architecture. En effet, chaque data center a son propre facteur d'émission qui, lui pourrait varier fortement d'un data center à un autre.

3.3.2 Algorithme génétique

Dans ce travail, nous avons commencé par élaborer un algorithme génétique capable de résoudre le problème VMPP. Cet algorithme étant un algorithme à population se base essentiellement sur les individus de chaque génération. Avant de fixer les différentes opérations de l'algorithme génétique, il faudra trouver la meilleure manière pour représenter ces individus. Ces derniers devront être élaborés de telle sorte que les opérations de l'algorithme génétique, principalement les opérations de croisement et de mutation seront facilement applicables.

Le problème de placement des machines virtuelles dans les data centers est un problème qui a pour objectif de trouver une localisation pour chaque VM dans les serveurs. De ce fait, une représentation de l'individu comme une liste des serveurs hébergeant les VMs pourrait être appliquée. Le premier élément de la liste représente la première machine virtuelle, le dernier élément de la liste représente la localisation de la dernière machine virtuelle. La figure 3.2 représente un exemple d'un individu dans le cas d'une solution à dix machines virtuelles.

VM1	VM2	VM3	VM4	VM5	VM6	VM7	VM8	VM9	VM10
5	6	4	3	7	7	2	1	6	1

Figure 3.1 Représentation d'un chromosome

Chaque élément du chromosome représente un gène. Ce gène désigne dans cet algorithme l'emplacement de la machine virtuelle et plus précisément l'identifiant du serveur qui l'héberge. Ainsi toutes les opérations seront appliquées sur l'ensemble des individus de la population.

Avant de commencer de les appliquer, il faudra débiter par initialiser les paramètres de l'algorithme. Ces paramètres sont le nombre d'individus dans la population initiale ainsi que le nombre d'itérations à réaliser représentant le nombre de générations à créer. Le choix de ces paramètres est arbitraire et se définira en fonction des tests initiaux.

L'algorithme génétique adopté dans ce travail est détaillé dans l'algorithme 1.

Algorithme 1 : Algorithme génétique

```

1: Entrée caractéristiques des VMs, informations sur les DCs,  $liste_{solution}$ 
2: Initialisation  $liste_{meilleursIndividus} := []$ ,  $p_{croisement} = 0.8$ ,  $p_{mutation} = 0.01$ 
3: Pour  $nb_{gen}$  de 1 à  $nb_{générations}$  faire
4:   Pour  $nb_p$  de 1 à  $nb_{populations}$  faire
5:      $liste_{selection} := []$ ,  $liste_{Res} := []$ 
6:     Pour  $n_i$  de 1 à  $nb_{individus_{gen}}$  faire
7:       Si  $réalisabilité(liste_{solution}[n_i]) == vrai$  alors
8:         Si  $coût(liste_{solution}[n_i]) \leq getMaxCostAllowedSolution()$  alors
9:            $liste_{meilleursIndividus} \cdot ajouterElément(solution_{n_i})$ 
10:        sinon  $liste_{selection} \cdot ajouterElement(solution_{n_i})$ 
11:      finsi
12:    finsi
13:     $liste_{croisement} := []$ 
14:    Pour  $i_p$  de 1 à  $nb_{individus} \div 2$  faire
15:       $(individu1, individu2) := choisirDeuxIndividus(liste_{selection})$ 
16:       $p_c = aléatoire(0,1)$ 
17:      Si  $p_c < p_{croisement}$  alors
18:         $op_{croisement} := aléatoire(1, nb_{individus_{gen}})$ 
19:         $(n_{i1}, n_{i2}) := croisement(individu1, individu2, op_{croisement})$ 
20:         $liste_{croisement} \cdot ajouterElément(n_{i1})$ 
21:         $liste_{croisement} \cdot ajouterElément(n_{i2})$ 
22:      sinon  $liste_{croisement} \cdot ajouterElément(individu1)$ 
23:       $liste_{croisement} \cdot ajouterElément(individu2)$ 
24:    finsi
25:  fin pour
26:  Pour  $i_p$  de 1 à  $nb_{individus}$  faire
27:     $prob_{mutation} := aléatoire(0,1)$ 
28:    si  $prob_{mutation} < p_{mutation}$  alors
29:       $liste_{Res} \cdot ajouterElément(mutation(liste_{croisement}[i_p]))$ 
30:    sinon  $liste_{Res} \cdot ajouterElément(liste_{croisement}[i_p])$ 
31:  finsi
32:  fin pour
33:   $liste_{solution} := combinaison(liste_{Res}, liste_{meilleursIndividus})$ 
34: fin pour
35: fin pour
36: fin pour

```

Cet algorithme montre les différentes opérations réalisées dans l'algorithme génétique afin de minimiser le coût de l'empreinte carbone. Tout d'abord, nous allons commencer par la sélection des solutions réalisables, ou parfois les solutions qui ne sont pas réalisables, mais qui peuvent l'être avec une légère modification. Parmi elles, nous choisissons un seuil du coût de la solution. Ce seuil-là nous permettra de faire passer les solutions ayant un coût inférieur à celui-ci directement à la prochaine génération sans passer par l'étape de croisement ni de mutation. Les autres solutions passeront par le croisement ainsi que la mutation. Cette étape-là s'appelle l'élitisme. Le nombre de solutions acceptées ne devrait pas être grand. Ainsi le seuil devra être fixé de manière à ce que peu de solutions passent directement à la population suivante.

Lors de l'opération de croisement, deux individus seront sélectionnés arbitrairement. Ainsi, les solutions ayant un coût assez proche seront choisies. Dès lors, un point de croisement déterminera le gène à partir duquel l'opération sera effectuée. Nous obtiendrons au final deux autres individus avec des coûts différents.

Sur la nouvelle population obtenue, nous effectuerons selon une probabilité faible la mutation qui consiste à changer l'emplacement d'une seule machine virtuelle. Les opérations de sélection, croisement et mutation se répèteront pour chaque individu et pour chaque population jusqu'à atteindre le nombre de générations fixé préalablement.

Au final, nous nous retrouverons avec une population à partir de laquelle nous déterminerons la meilleure solution obtenue par l'algorithme génétique.

3.3.3 Algorithme mémétique

L'algorithme mémétique dans ce travail sera la combinaison entre l'algorithme génétique détaillé dans la section précédente avec l'algorithme de recherche tabou.

Étant la combinaison entre les deux métaheuristiques, il réunit la force de ces deux derniers. L'algorithme de recherche tabou s'appliquera sur chaque individu après les différentes opérations de l'algorithme génétique.

L'algorithme génétique utilisé sera le même que cité précédemment. Ainsi nous détaillerons l'algorithme de recherche tabou qui lui sera appliqué après la dernière opération de l'algorithme génétique à savoir la mutation et donc après la ligne 35 de l'algorithme 1.

L'algorithme de recherche tabou est présenté dans l'algorithme 2.

Algorithme 2 : Algorithme de recherche tabou

```

1: Initialisation  $liste_{tabou} := []$  , solution initiale  $S$ 
2:  $nb_{itération} := max_{itération}$  ,  $iter := 0$ 
3: tant que  $iter < nb_{itération}$  faire
4:   Pour  $n_s$  de 1 à  $nb_{vm}$  faire
5:      $mvt := effectuerMouvement (individu[n_s])$ 
6:     si  $(coût(S') < coût(S) \ \&\& \ accepterMouvement(mvt) == vrai)$ 
7:        $liste_{tabou}.ajouterElement(mvt)$ 
8:        $S := S'$ 
9:   finsi
10:  fin pour
11:   $iter ++$ ;
12: fin tant que

```

Au départ chaque individu dans l'algorithme génétique représente la solution initiale pour l'algorithme de recherche tabou. Ainsi nous démarrerons par la configuration donnée par cet individu.

L'algorithme de recherche tabou, appliqué sur les individus générés par les opérations de l'algorithme génétique, commencera par des mouvements qui ne sont autres que des réaffectations des machines virtuelles dans de nouveaux serveurs. Le voisinage étant formé par les diverses localisations sera visité pour définir le meilleur mouvement à appliquer. La caractérisation du meilleur mouvement se fera sur un ensemble de tests que nous évaluerons au fur et à mesure. Le choix de la meilleure solution sera fait sur l'ensemble du voisinage. Chaque solution acceptée sera insérée dans une liste taboue.

Afin de ne pas retomber dans une solution déjà élaborée, nous vérifierons si cette dernière appartient à la liste taboue que mettons à jour à chaque fois qu'une solution a été acceptée.

Une fois que le nombre d'itérations fixé a été atteint, nous arrêterons le déroulement de l'algorithme et passerons au prochain individu. Pour chaque génération, et après avoir appliqué les opérations de l'algorithme génétique, nous exécuterons l'algorithme de recherche tabou.

Afin que nous puissions obtenir de meilleurs résultats et que l'algorithme TS puisse visiter le maximum de solutions, nous appliquerons en premier lieu la mémoire à court terme, puis la mémoire à moyen terme et enfin la mémoire à long terme.

Lors de la mémoire à court terme, nous vérifierons à chaque fois si la nouvelle solution a été obtenue suite à un mouvement tabou ou pas. Cette vérification se fait au moyen de la liste taboue. Si le mouvement n'est pas considéré comme tabou, nous acceptons la solution. Dans le cas contraire, un critère d'aspiration pourrait permettre d'accepter la solution s'il améliore la solution courante.

La mémoire à moyen terme dans la TS représente l'étape d'intensification alors que la mémoire à long terme, elle, représente la diversification.

Le mécanisme de mémoire à moyen terme permettra un retour à la meilleure solution obtenue à un moment donné, afin de relancer l'algorithme de TS à partir de celle-ci. L'intensification a pour objectif d'adopter d'autres directions de recherche des nouvelles solutions.

Lors de l'utilisation de la TS avec l'algorithme génétique, plusieurs solutions initiales seront utilisées, étant donné le nombre important d'individus. Ainsi, le mécanisme de mémoire à long terme dans l'algorithme mémétique pourrait renfoncer l'algorithme de TS en sauvegardant les meilleures configurations obtenues tout au long de notre recherche afin de donner la chance à l'apparition de nouvelles configurations, d'où la diversification.

3.3.4 Recuit simulé

Afin que l'utilisation du recuit simulé dans ce travail soit le plus adéquate possible, il est important de faire un bon choix de paramètres. L'un des paramètres les plus importants est le facteur de température. Ce facteur se déduit à partir d'un nombre donné de perturbations d'une solution donnée. En effet, c'est la moyenne de la variation des solutions obtenues ΔE . Voulant débiter avec une bonne qualité de solution, nous devons choisir une température relativement basse T_0 . Le calcul de la température initiale sera donc déduit à partir de l'équation suivante : $T_0 = - \Delta E / \log(\theta)$

Le paramètre θ est un pourcentage qui interviendra dans la diminution de la température.

De plus et parmi les paramètres à fixer, il y a le nombre de paliers où, pour chacun de ces derniers, la température diminue avec un taux de décroissance choisi arbitrairement.

La taille de chaque palier est définie en fonction du nombre de mouvements n'améliorant pas la fonction objectif.

L'algorithme de recuit simulé est détaillé dans l'algorithme 3.

Algorithme 3 : Algorithme de recuit simulé

```

1: Initialisation configuration initiale  $S_1$  ; température initiale  $T$ 
2: nombre de mouvements non acceptés par palier  $nb_{mvtPal}$ 
3:  $E_1 := coût(S_1)$ 
4:  $iter := 0$  ;  $taux_{dec} := 0.99$  ;  $pal := 0$ 
5: tant que  $pal < nb_{mvtPal}$  faire
6:   pour  $iter$  de 0 à  $nb_{itérations}$  faire
7:      $mvt := effectuerMouvement (vm, server)$ 
8:      $E_2 := coût(S_2)$ 
9:     si ( $coût(S_2) \leq coût(S_1)$ 
10:       &&  $accepterMouvement(mvt) == vrai$ ) alors
11:        $validerMouvement(mvt)$ 
12:        $S_1 := S_2$ 
13:        $E_1 = coût(S_1)$ 
14:        $pal := 0$ 
15:       Sinon  $accepterMouvement(mvt)$  avec probabilité  $= exp(-\Delta E/T)$ 
16:        $E_1 = coût(S_2)$ 
17:        $pal := 0$ 
18:       Sinon  $pal++$ 
19:     finsi
20:    $iter++$ 
21: fin pour
22: fin tant que

```

CHAPITRE 4 IMPLÉMENTATION ET ANALYSE DES RÉSULTATS

Dans ce chapitre, nous présenterons les différentes expérimentations élaborées avec les algorithmes de métaheuristique. Nous débuterons par présenter les jeux de données, ensuite les paramètres utilisés pour les différents algorithmes utilisés ainsi que les résultats obtenus. Puis, nous présenterons l'influence de chaque paramètre du data center sur l'empreinte carbone. Enfin, nous évaluerons les algorithmes utilisés.

4.1 Paramètres d'implémentation

Afin de créer un environnement capable de simuler un cas réel, nous avons besoin de spécifier les diverses caractéristiques des différents éléments constituant d'un côté le data center et aussi les machines virtuelles ainsi que leurs liaisons entre elles.

A partir de la topologie d'un data center, nous pouvons connaître les caractéristiques de chaque composant comment indiqué dans le tableau 4.1.

Tableau 4.1 Caractéristiques des composants de l'environnement

Composants	Caractéristiques
Data center	<ul style="list-style-type: none"> - Facteur d'émission - Nombre de châssis - Nombre de serveurs
Châssis	<ul style="list-style-type: none"> - Puissance initiale - Nombre de serveurs - Nombre de ventilateurs - Maximum température dégagée - Résistance thermique
Commutateur	<ul style="list-style-type: none"> - Puissance initiale - Puissance du trafic - Capacité de bande passante
Serveur	<ul style="list-style-type: none"> - Puissance initiale - Capacité du processeur - Taille de mémoire - Capacité du disque - Puissance du processeur - Puissance du disque

Comme présenté dans la revue de littérature, la topologie d'un data center, prenant en considération les différents commutateurs garantissant le trafic intra-data centers, les châssis ainsi que les serveurs, doit être totalement détaillée dans les fichiers de données.

Le tableau 4.2 détaille les caractéristiques des machines virtuelles présentées dans un second fichier de données.

Tableau 4.2 Caractéristiques des VMs

Composant	Caractéristiques
Machine Virtuelle	<ul style="list-style-type: none"> - Capacité du processeur - Taille de mémoire - Capacité du disque - VMs en liaison - Propriétaire de la VM

À travers les différentes caractéristiques des composants d'un data center ainsi que celles des machines virtuelles présentées dans les différents fichiers d'entrée, nous serons en mesure d'exécuter l'étape de placement des VMs dans l'environnement infonuagique.

4.2 Implémentation des algorithmes de métaheuristique

Avant de débiter le déroulement des algorithmes de métaheuristique, nous fixerons les paramètres de chacun d'eux. Le choix de ces paramètres est important pour assurer la réussite du fonctionnement des algorithmes. Nous allons donc présenter dans cette partie les paramètres associés à l'algorithme génétique, l'algorithme mémétique ainsi que le recuit simulé puis les résultats obtenus pour chacun d'eux.

4.2.1 Implémentation de l'algorithme génétique

Cette section sera consacrée à la présentation de l'algorithme génétique en commençant par la définition du paramètre associé à GA puis présenter les résultats obtenus.

Paramètres associés à l'algorithme génétique

L'algorithme génétique est un algorithme à population. Pour sa bonne exécution, il faudra commencer par fixer quelques paramètres à savoir la taille de la population initiale avec laquelle

l'AG débutera ainsi que le nombre d'itérations. Le nombre d'itérations qui correspond aux nombres de générations créées sera choisi arbitrairement. Dans notre travail il sera égal à 150 itérations. Pour le deuxième paramètre, plusieurs tests devront être élaborés afin de retenir la meilleure valeur pour celui-ci.

Afin de fixer ce paramètre, nous avons effectué plusieurs tests, en prenant à chaque fois plusieurs valeurs différentes, élaborés sur trois instances différentes et comparés les résultats obtenus.

Le tableau 4.3 présente les trois instances utilisées.

Tableau 4.3 Instances utilisées pour la définition du paramètre de l'AG

Instances	Nombre de data centers	Nombre de commutateurs	Nombre de châssis	Nombre de serveurs	Nombre de VMS
1	2	18	8	24	60
2	2	30	20	100	500
3	6	90	60	300	770

La figure 4.1 illustre les différents résultats obtenus avec diverses instances caractérisées chacune par son nombre d'individus dans la population initiale.

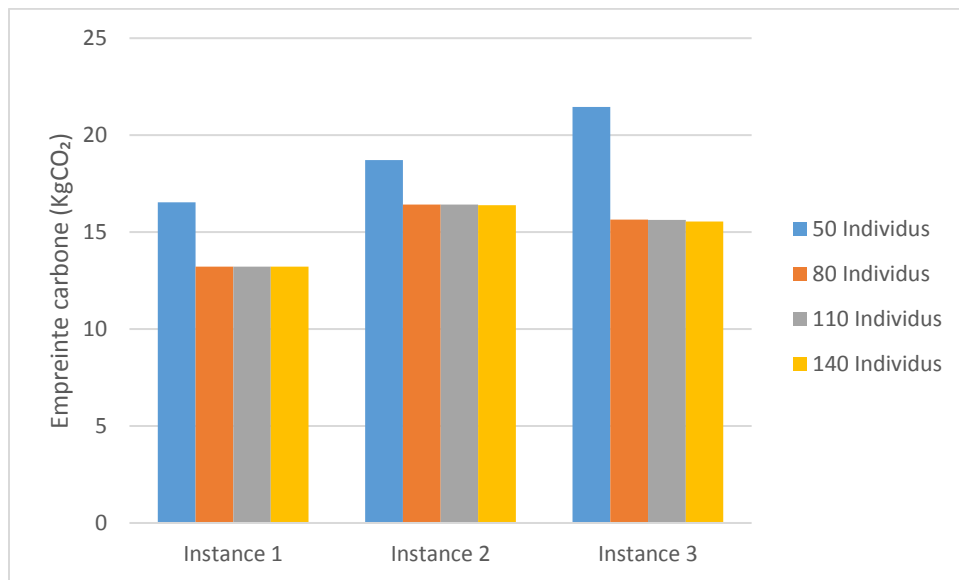


Figure 4.1 Choix du paramètre du nombre d'individus de la population initiale

À partir de cette figure, nous remarquons que les différentes tailles de la population initiale utilisées correspondant à 80, 110 et 140 individus, donnent des résultats assez proches. Tandis l'utilisation de 50 individus pour la population initiale présente une empreinte carbone relativement élevée. Ainsi pour départager ces trois configurations nous avons comparé leur temps d'exécution. La figure 4.2 illustre le temps d'exécution des trois instances pour des tests réalisés sur 150 itérations.

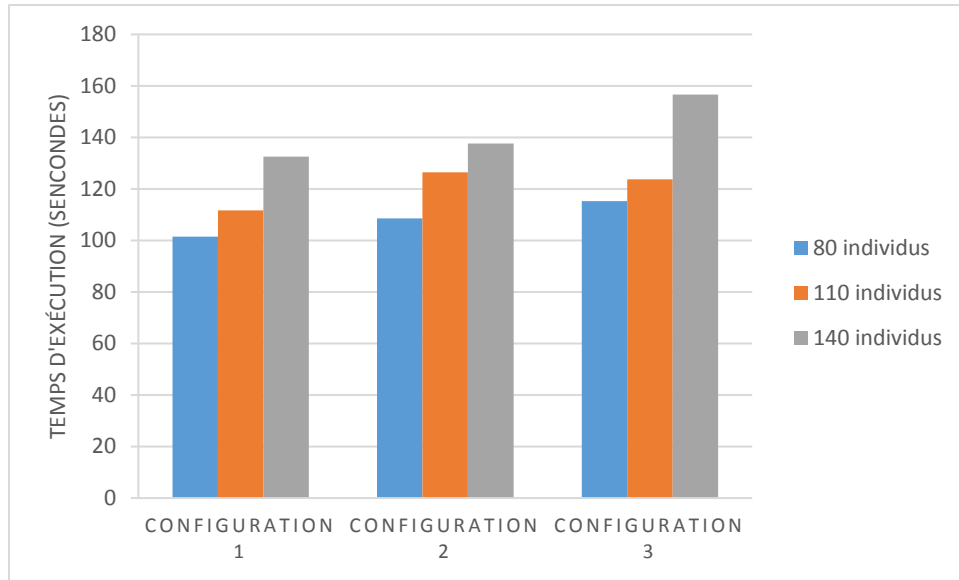


Figure 4.2 Comparaison du temps d'exécution entre les trois configurations

Ainsi, nous remarquons que le temps d'exécution pour une population initiale de 80 individus est considérablement faible comparé aux solutions démarrées par une population de 110 individus et 140 individus. Ainsi, le choix du paramètre de la population initiale, qui est égale à 80, est justifié.

Résultats obtenus avec l'algorithme génétique

En utilisant le paramètre comme indiqué précédemment, l'algorithme génétique pour le problème de placement des machines virtuelles permet de présenter des résultats obtenus pour la minimisation de l'empreinte carbone. Plusieurs tests ont été élaborés pour diverses instances et pour une population d'individus de taille 80.

Le tableau 4.4 détaille les résultats obtenus par l'algorithme génétique pour le VMPP.

Chaque instance est caractérisée par un nombre de data centers noté D, commutateurs noté T, châssis noté X, serveurs noté S, machines virtuelles noté V.

Tout d'abord, la performance de l'algorithme génétique sera calculée en termes de coût d'empreinte carbone. Ensuite, nous devons prendre en considération le nombre de serveurs inactifs avec laquelle la meilleure solution a été trouvée. Puis le temps d'exécution qui dans ce cas-là a été pris pour finir les 150 itérations. Enfin le numéro de l'itération qui a donné le meilleur résultat.

Tableau 4.4 Résultats détaillés de l'algorithme génétique

Inst.	(D, T, X, S, V)	Emp. Carb. (t)	Nb serveurs inactifs (%)	Temps d'exécution (s)	Num. itération
1	1, 11, 6, 18, 60	4.88	16	108.19	23
2	1, 15, 10, 50, 60	3.47	24	107.46	33
3	1, 15, 10, 50, 430	11,18	14	175.12	29
4	2, 22, 12, 60, 60	3.08	28	99.13	30
5	2, 30, 20, 100, 60	3.03	25	116.57	14
6	2, 30, 20, 100, 500	10,45	34	189.26	35
7	3, 33, 18, 54, 60	2.64	54	136.27	12
8	3, 33, 18, 54, 52	2.76	39	129.89	17
9	3, 33, 18, 90, 46	24,68	70	134.37	21
10	4, 44, 24, 72, 51	16,42	53	146.15	28
11	4, 60, 40, 200, 580	13,42	25	156.75	39
12	4, 90, 60, 300, 500	15,26	25	184.58	38

À partir du tableau précédent, nous pourrions déduire le comportement de l'algorithme génétique. La majeure partie des meilleures solutions retrouvées a été découverte lors des premières itérations. En effet, il a suffi de moins de 40 itérations pour réussir à trouver un bon placement des machines virtuelles dans les data centers. De plus, le temps d'exécution de cet algorithme pour finir toutes les itérations ne dépasse pas les 200 secondes pour les petites comme pour les grandes instances.

4.2.2 Implémentation de l'algorithme mémétique

L'algorithme mémétique est un algorithme d'optimisation hybride. Nous allons devoir donc essayer de fixer les paramètres des deux algorithmes qui constituent cette hybridation à savoir l'algorithme génétique ainsi que l'algorithme de recherche tabou. Nous montrerons ensuite les résultats obtenus par ce même algorithme.

Paramètres associés à l'algorithme mémétique

L'algorithme mémétique étant une combinaison entre l'algorithme génétique et l'algorithme de recherche tabou, nous considérerons le même nombre d'individus utilisé dans GA pour la population initiale.

Pour l'algorithme de recherche tabou, le choix de la taille de la liste tabou dépendra de chaque configuration étant donné que ces dernières n'ont pas toutes les mêmes caractéristiques. De plus, la taille de l'espace de recherche influencera directement sur ce paramètre.

Si dans une configuration donnée, plusieurs data centers disposent d'un espace libre pour l'hébergement d'une ou plusieurs machines virtuelles alors l'espace de recherche augmentera et les mouvements aussi suivront. Dans ce cas, plus le nombre de possibilités de mouvements augmente et plus la taille de la liste taboue augmente afin de limiter au maximum les mouvements tabous.

La liste taboue retient des informations concernant les derniers mouvements tabous. Ces informations englobent la machine virtuelle déplacée, l'ancien serveur qui l'a hébergée ainsi que le nouveau serveur.

Afin de choisir la bonne taille de la liste taboue, plusieurs tests doivent être élaborés. Dans cette section, nous avons choisi 6 configurations pour montrer l'efficacité de la taille de la liste taboue sur l'obtention de la meilleure solution. Le tableau 4.4 donne des informations concernant les 6 instances utilisées.

Tableau 4.5 Instances utilisées pour le choix de la taille de la liste taboue

Instances	Nombre de data centers	Nombre de commutateurs	Nombre de châssis	Nombre de serveurs	Nombre de VMS
1	1	9	4	12	56
2	1	9	4	16	56
3	2	18	8	24	60
4	2	30	20	100	500
5	6	90	60	300	770
6	8	120	120	400	660

La figure 4.3 illustre la différence dans l'efficacité de la taille de la liste taboue entre les six instances utilisées.

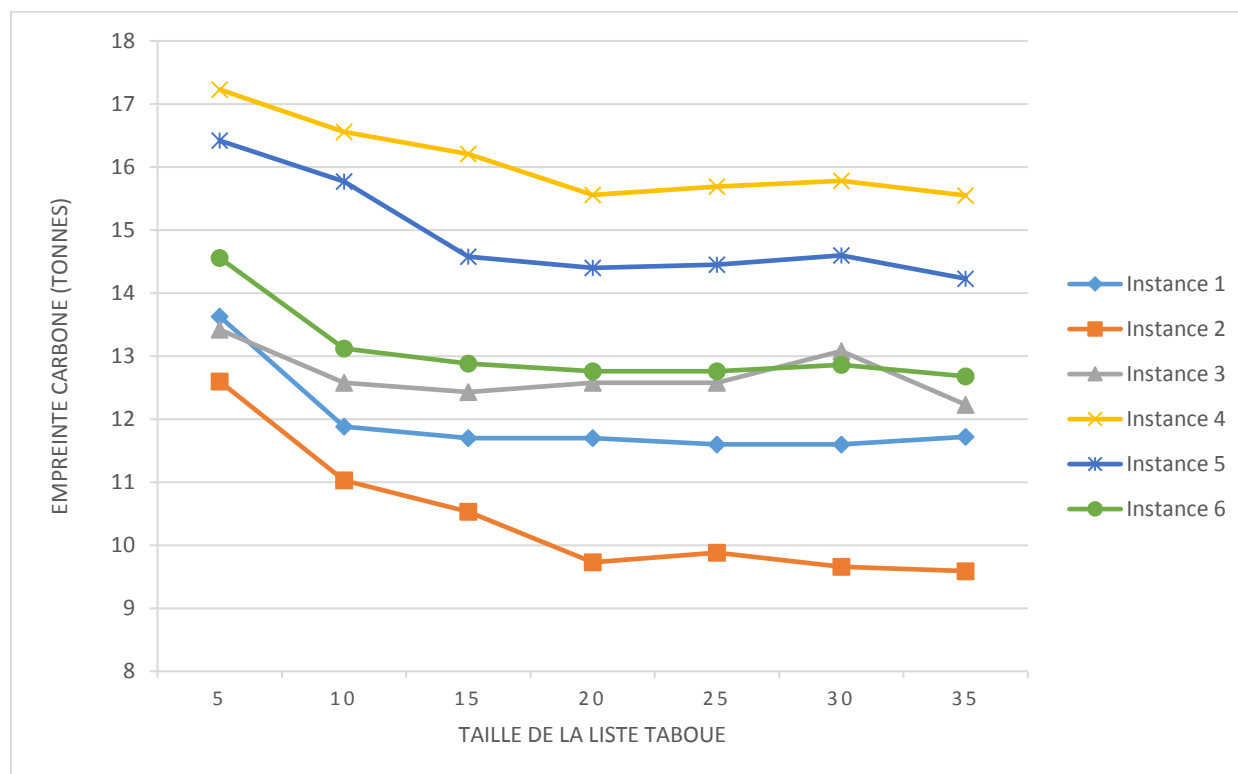


Figure 4.3 Influence de la taille de la liste taboue sur la solution obtenue

D'après la figure 4.3, pour les six instances utilisées, nous constatons que quand la taille de la liste taboue dépasse 20, nous obtenons les meilleurs résultats.

De ce fait, pour chaque instance, multiples tests seront élaborés avec cette même taille. Nous garderons ainsi le meilleur résultat obtenu.

Résultats obtenus avec l'algorithme mémétique

Une fois que les paramètres de l'algorithme mémétique sont fixés, nous commencerons par réaliser les tests sur les mêmes jeux de données utilisés pour l'algorithme génétique.

Le tableau 4.5 présente les résultats obtenus par MA. Tout comme pour l'algorithme génétique, la performance de l'algorithme mémétique sera calculée principalement en termes de coût d'empreinte carbone. Ce tableau démontre que les meilleures solutions ont été obtenues vers la fin

de l'exécution de l'algorithme. L'exploration de nouvelles solutions continue donc tout au long de l'exécution du MA.

Tableau 4.6 Résultats détaillés de l'algorithme mémétique

Inst.	(D, T, X, S, V)	Emp. Carb. (t)	Nb serveurs inactifs (%)	Temps d'exécution (s)	Num. itération
1	1, 11, 6, 18, 60	4.5	19	126.81	66
2	1, 15, 10, 50, 60	3.38	26	112.57	67
3	1, 15, 10, 50, 430	10,04	18	263.89	55
4	2, 22, 12, 60, 60	3.04	28	211.49	41
5	2, 30, 20, 100, 60	3.02	26	217.18	72
6	2, 30, 20, 100, 500	9,8	37	401.36	84
7	3, 33, 18, 54, 60	2.63	55	300.27	14
8	3, 33, 18, 54, 52	2.74	37	307.46	135
9	3, 33, 18, 90, 46	21,41	73	352.35	96
10	4, 44, 24, 72, 51	14,4	64	409.12	66
11	4, 60, 40, 200, 580	12,76	37	457.46	94
12	4, 90, 60, 300, 500	14,27	37	444.53	103

Le nombre de serveurs inactifs, contrairement à l'algorithme génétique, présente un pourcentage assez élevé. Cela s'explique par le fait que réduire l'empreinte carbone revient aussi à réduire le nombre de serveurs actifs dans les data centers.

Ce même tableau démontre aussi que contrairement à l'algorithme génétique, l'algorithme mémétique prend plus de temps pour son exécution. En effet, et dans chaque itération, nous exécutons l'algorithme de recherche tabou sur chaque chromosome. Chacun de ces derniers représente la disposition des machines virtuelles dans les data centers et donc sa taille est relative au nombre de VMs. Augmenter le nombre de machines virtuelles ralentit donc l'exécution de l'algorithme mémétique.

L'empreinte carbone obtenue avec l'algorithme mémétique est considérablement moins élevée que celle obtenue avec l'algorithme génétique, sauf que pour le MA nous mettons plus de temps à retrouver la meilleure solution.

4.2.3 Implémentation du recuit simulé

Le recuit simulé est un algorithme qui s'adapte facilement au problème de placement des machines virtuelles dans un environnement infonuagique. Le seul inconvénient serait de faire le bon choix des paramètres. Nous aurons donc à justifier le choix des paramètres puis présenter les résultats obtenus avec cet algorithme.

Paramètres associés au recuit simulé

Le premier paramètre à choisir est la température initiale. Ce paramètre sera déduit à partir de plusieurs solutions. En appliquant une centaine de perturbations, nous serons en mesure de calculer la moyenne des différences entre les coûts d'empreinte carbone de toutes ces perturbations.

Ainsi la température initiale sera le résultat de ces diverses perturbations. Ensuite afin de pouvoir démarrer avec une qualité de solution plutôt bonne, nous ajouterons un facteur θ . Plus ce facteur est proche de zéro et plus la qualité sera meilleure. Pour le VMPP, θ sera pris égale à 0.2. Par conséquent, la température initiale sera égale à 140.

Une fois que la température initiale fixée, nous devrions maintenant fixer la loi de décroissance de la température. Ce paramètre-là est dans la plupart des cas proche de 1. Dans ce travail il prendra la valeur de 0.99. Ainsi la loi de décroissance sera $T_{k+1} = 0.99 T_k$.

La température pour chaque palier diminuera selon le paramètre précédemment fixé. Pour cela, il faudra définir la longueur du palier. La longueur du palier correspond au nombre de configurations visitées pour un palier donné.

Résultats obtenus avec le recuit simulé

Les paramètres étant fixés, nous présenterons les résultats obtenus avec le recuit simulé. Le coût d'empreinte carbone est toujours la principale manière pour évaluer la performance de l'algorithme.

Le tableau 4.6 présente les résultats obtenus avec le recuit simulé pour le problème de placement des machines virtuelles dans les data centers. Ce tableau démontre la rapidité d'exécution du recuit simulé comparé à l'algorithme génétique et l'algorithme mémétique. Pour un nombre d'itérations égal au nombre de générations créées pour les deux précédents algorithmes, nous constatons que temps d'exécution pour le recuit simulé est relativement faible.

Trouver une valeur d’empreinte carbone un peu plus élevée que celle obtenue par l’algorithme mémétique, mais assez proche en si peu de temps rend l’algorithme de recuit simulé intéressant pour les grandes instances contenant un nombre important de machines virtuelles.

Tableau 4.7 Résultats détaillés du recuit simulé

Inst.	(D, T, X, S, V)	Emp. Carb. (t)	Nb serveurs inactifs (%)	Temps d’exécution (s)	Num. itération
1	1, 11, 6, 18, 60	4.73	16	135.23	115
2	1, 15, 10, 50, 60	3.32	24	148.25	92
3	1, 15, 10, 50, 430	10,36	15	265.34	75
4	2, 22, 12, 60, 60	3.05	28	124.68	100
5	2, 30, 20, 100, 60	2.84	21	106.73	59
6	2, 30, 20, 100, 500	10,09	36	244.23	71
7	3, 33, 18, 54, 60	2.63	58	168.15	82
8	3, 33, 18, 54, 52	2.76	36	158.83	73
9	3, 33, 18, 90, 46	21,41	68	101.56	39
10	4, 44, 24, 72, 51	14,98	59	106.89	59
11	4, 60, 40, 200, 580	18,03	29	211.82	95
12	4, 90, 60, 300, 500	14,84	25	199.33	53

Contrairement à l’algorithme mémétique, le recuit simulé n’est pas autant affecté par l’augmentation du nombre de VMs. En effet, le temps d’exécution pour les grandes instances n’est pas aussi élevé et pourrait être considéré comme rapide pour la recherche de la meilleure solution.

Le nombre de serveurs actifs, quant à lui, est, dans la plupart des instances, inférieur à celui de l’algorithme mémétique. C’est tout à fait évident, car avec l’algorithme mémétique nous avons tendance à visiter plus de solutions et donc plus de possibilité de combinaisons avec les serveurs qui hébergent les VMs.

Avec le recuit simulé, nous avons tendance à trouver la meilleure solution dans les dernières itérations. En effet, étant donné que la probabilité d’acceptation des moins bonnes solutions diminue au fur et à mesure que l’algorithme avance.

4.3 Influences des paramètres de l’environnement

Le calcul de la fonction objectif varie en fonction du nombre de composants qui existent dans l’environnement infonuagique. Le nombre de data centers, commutateurs, châssis, serveur,

machines virtuelles et lien entre les VMs sont principalement les paramètres qui participent à la variation du coût total d’empreinte carbone. Dans cette section nous allons varier chaque paramètre et spécifier lequel qui participe le plus dans cette variation.

4.3.1 Nombre de data centers

Dans un environnement infonuagique, nous pourrions trouver un ou plusieurs data centers qui hébergeront tous les équipements. Lors du placement des machines virtuelles dans les DCs nous devons prendre en compte le nombre de data centers dans l’architecture globale, car le fait de disperser les VMs les unes des autres influe directement sur l’empreinte carbone. Ainsi pour un même nombre de VMs fixé, nous varierons le nombre de data center et découvrirons comment varie la valeur de l’empreinte carbone dans cet environnement.

Les configurations choisies seront donc les mêmes pour tous les tests, à l’exception du nombre de data centers, qui lui sera la seule variable. Les data centers ajoutés auront le même besoin en énergie pour l’hébergement du même nombre de machines virtuelles. Pour la comparaison, nous avons réalisé 5 tests avec 1, 2, 4, 6 puis 8 data centers. La figure 4.4 illustre l’influence du nombre de DCs sur l’empreinte carbone.

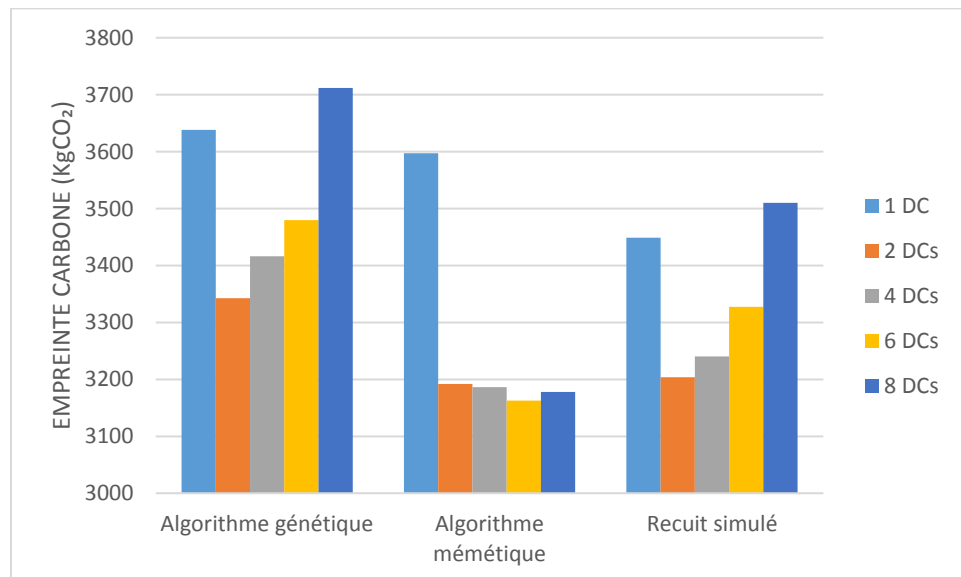


Figure 4.4 Influence de la variation du nombre de data centers sur l'empreinte carbone

Ainsi, nous remarquons que pour les instances à un seul data center, l'empreinte carbone est relativement élevée étant donné que les serveurs seront plus saturés nécessitant ainsi plus d'énergie pour leurs refroidissements.

L'algorithme génétique et le recuit simulé ont un comportement identique face à la variation du nombre de data centers. Néanmoins, l'empreinte carbone résultante du SA reste moins élevée que celle avec l'algorithme génétique.

Contrairement aux deux derniers, l'algorithme mémétique se comporte plutôt bien. En effet, grâce à l'algorithme de recherche tabou, nous avons pu garder pratiquement les mêmes valeurs d'empreinte carbone pour 2, 4, 6 et 8 data centers. C'est l'intensification apportée par le TS qui a permis cela.

4.3.2 Nombre de commutateurs

Maintenant que nous savons comment chaque algorithme se comporte avec la variation du nombre de data centers, nous allons varier le nombre de commutateurs et connaître si les trois algorithmes utilisés ont le même comportement.

La figure 4.5 illustre l'influence de la variation du nombre de châssis sur l'empreinte carbone.

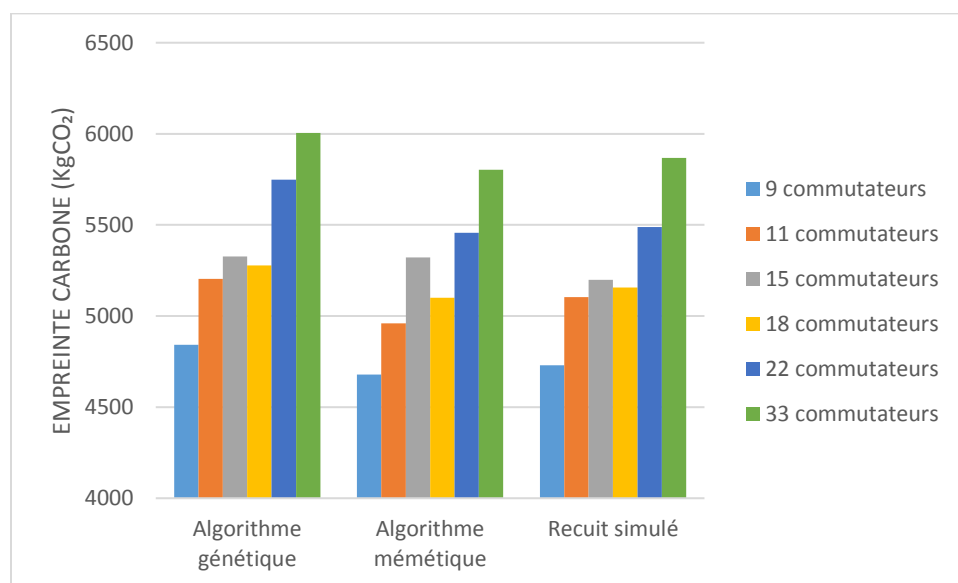


Figure 4.5 Influence de la variation du nombre de commutateurs sur l'empreinte carbone

D'après la figure 4.5, nous constatons que si le nombre de commutateurs augmente, alors cela impactera sur l'empreinte carbone qui lui augmentera. Ceci s'explique par le fait que les machines virtuelles complémentaires auront tendance à passer par plus de commutateurs pour communiquer.

Le trafic réseau sera plus chargé et par conséquent nous aurons besoin de plus d'énergie pour assurer une communication inter-VMs parfaite et donc une bonne qualité de service.

4.3.3 Nombre de châssis

Les châssis sont les composantes qui hébergeront les serveurs au sein d'un data center. Ils sont responsables pour le refroidissement des serveurs ainsi que l'alimentation électrique requise pour leur fonctionnement.

Nous essayerons de connaître l'influence du nombre de châssis sur l'empreinte carbone en variant ce nombre-là et tester avec les différents algorithmes.

La figure 4.6 illustre le comportement des trois algorithmes à savoir, l'algorithme génétique, l'algorithme mémétique ainsi que le recuit simulé face à la variation du nombre de châssis

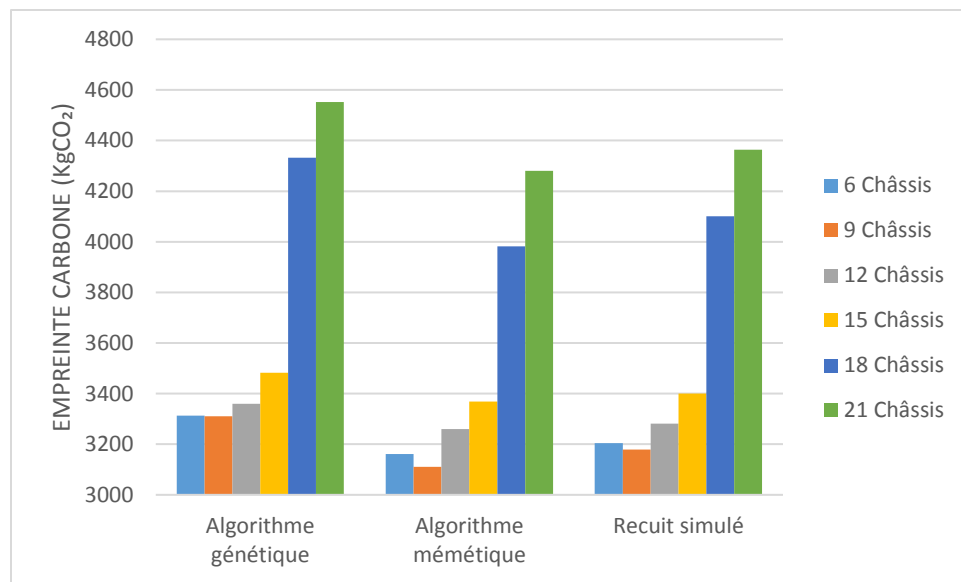


Figure 4.6 Influence de la variation du nombre de châssis sur l'empreinte carbone

On constate que pour les trois algorithmes, plus le nombre de châssis augmente et plus l'empreinte carbone augmente. Cela s'explique par le fait que les serveurs seront dispersés sur plusieurs châssis.

Par conséquent les VMs auront tendance à être éparpillés sur plusieurs serveurs appartenant à des châssis différents. Tout ceci affectera le trafic réseau qui lui sera dense et donc l’empreinte carbone qui augmente.

4.3.4 Nombre de serveurs

Les serveurs qui sont les composants responsables de l’hébergement des machines virtuelles assurent de fournir les ressources nécessaires aux VMs tout en s’assurant de laisser une certaine marge au niveau des ressources pour garantir une élasticité lorsque ces VMs en demandent plus.

Pour le problème de placement des machines virtuelles dans les data centers, augmenter le nombre de serveurs pourrait influencer sur le temps de recherche de la meilleure solution. En effet, nous aurons de plus de solution à visiter et de combinaison à tester pour savoir si cette dernière est la meilleure ou pas.

La figure 4.7 illustre l’influence de la variation de l’empreinte carbone en fonction du nombre de serveurs.

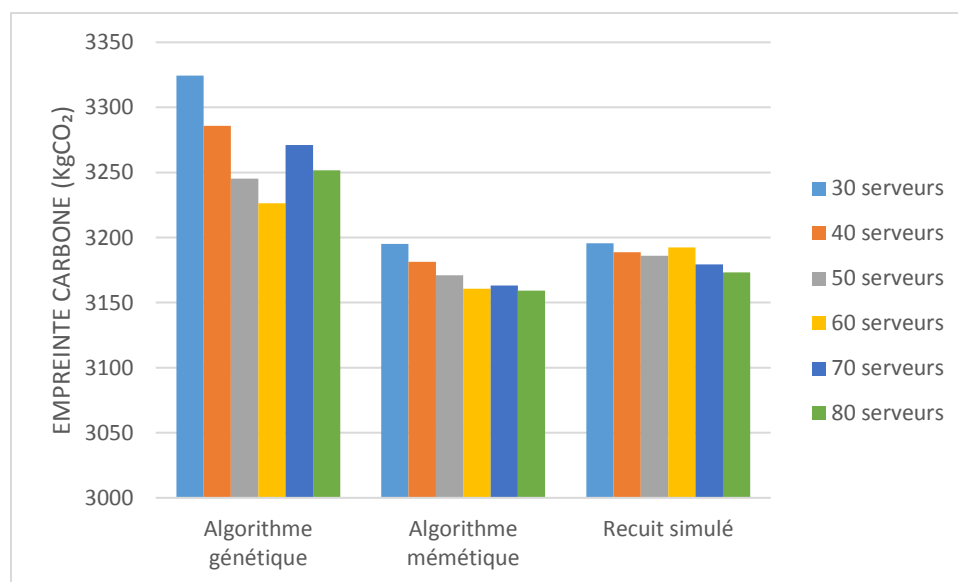


Figure 4.7 Influence de la variation du nombre de serveurs sur l’empreinte carbone

Nous constatons que lorsque le nombre de serveurs augmente, l’empreinte carbone diminue. Cela s’explique par le fait que plus le nombre de serveurs augmente et plus le nombre de serveurs accueillis par les châssis augmente. Par conséquent, nous sommes en mesure de réunir un nombre

considérable de machines virtuelles au niveau du même châssis. Tout cela engendrera un trafic moins dense puisque tout se passe au niveau du même châssis. D'autant plus que maintenir une température idéale au niveau du même composant, qui dans ce cas-là est le châssis, engendrera une consommation d'énergie moins importante que si nous avons affaire à gérer plus de châssis.

Augmenter le nombre serveurs pour chaque châssis participera à l'augmentation du nombre de châssis passifs où tous les serveurs qu'ils contiennent n'hébergent aucune machine virtuelle. Ainsi diminuer le nombre de châssis passifs participera à la diminution de l'empreinte carbone.

4.3.5 Nombre de machines virtuelles

Les machines virtuelles dans le problème de placement des machines virtuelles dans un environnement infonuagique constituent le composant le plus variable parmi tous les autres composants. En effet, tout le problème consiste à trouver, en fonction des caractéristiques de chacune des VMs, une bonne gestion des ressources de l'environnement.

Tenir compte des ressources disponibles au niveau de chaque serveur ainsi que des liaisons entre les VMs nous chercherons la meilleure approche pour placer ces VMs-là.

La figure 4.8 illustre le comportement de l'algorithme génétique, l'algorithme mémétique ainsi que le recuit simulé face à la variation du nombre de machines virtuelles sans augmentation du nombre de liaisons entre les VMs. Ainsi toutes les VMs ajoutées seront indépendantes.

D'après la figure 4.8, augmenter le nombre de VMs implique l'augmentation de l'empreinte carbone. Ce résultat est prévisible. En effet, chaque machine virtuelle ajoutée nécessite des ressources que les serveurs devront satisfaire. Ceci va, par conséquent, participer à l'augmentation de la charge de travail des serveurs qui, forcément, va augmenter la température interne. Si les serveurs surchauffent, la qualité de service proposée par le data center en question se verra détériorer.

Afin de garder une bonne QoS, le refroidissement du serveur est obligatoire, mais cela aura des impacts sur l'empreinte carbone qui augmentera.

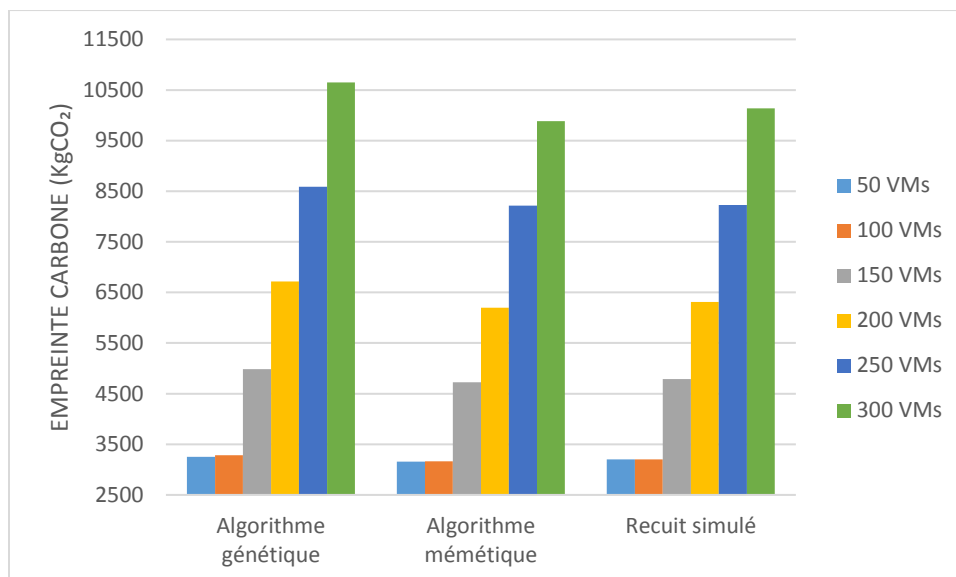


Figure 4.8 Influence de la variation du nombre de machines virtuelles sur l'empreinte carbone

4.3.6 Nombre de liaisons entre les machines virtuelles

Un client qui souhaite que ses machines virtuelles soient hébergées dans un data center externe, impose généralement quelques conditions. En effet, si ses VMs nécessitent une communication entre elles, le fournisseur doit la prendre en considération.

Le trafic global au sein d'un data center est principalement constitué des échanges inter-VMs à l'intérieur des DCs. Réduire le trafic réseau revient à réduire la distance entre les machines virtuelles liées.

Dans ce travail, nous représentons la liaison entre une paire de machines virtuelles comme étant un flux qui circulera au sein du data center.

La figure 4.9 illustre la variation de l'empreinte carbone en fonction du flux.

D'après la figure précédente, nous constatons que, pour les trois algorithmes utilisés, si le flux augmente, alors l'empreinte carbone augmente. En effet, lorsque nous ajoutons des liaisons entre les différentes machines virtuelles au sein d'un data center, le trafic réseau intra-data centers sera plus dense, nécessitant par conséquent un refroidissement des composants et une énergie supplémentaire pour le maintien de la bonne qualité de service.

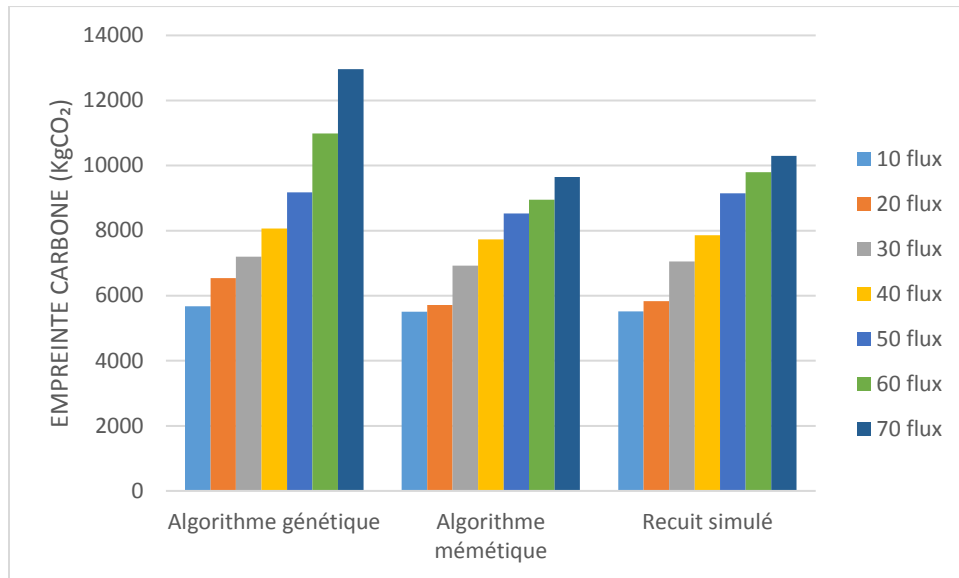


Figure 4.9 Influence de la variation du flux sur l'empreinte carbone

4.4 Évaluation des algorithmes utilisés

Afin d'évaluer les résultats obtenus avec l'algorithme génétique, l'algorithme mémétique ainsi que le recuit simulé, nous avons besoin de le comparer avec d'autres approches de résolution.

Dans ce travail figurent deux types d'instances en fonction de la taille du problème. D'abord, nous comparerons les résultats obtenus pour les petites instances avec la méthode exacte ainsi que l'algorithme de recherche tabou itéré.

Ensuite, nous comparerons les résultats obtenus pour les grandes instances avec seulement l'algorithme de recherche tabou itéré.

Chaque instance est caractérisée par un nombre de data centers noté D , commutateurs noté T , châssis noté X , serveurs noté S , machines virtuelles noté V et le flux noté F .

Le tableau 4.8 montre les résultats obtenus avec les différents algorithmes pour pouvoir ainsi les comparer avec la méthode exacte ainsi que l'algorithme de recherche tabou itéré.

Tableau 4.8 Comparaison entre la méthode exacte, ITS, GA, MA et SA pour les instances de petite taille

Inst.	(D, T, X, S, V, F)	Méth. exacte	ITS	GA	MA	SA
		Emp. Carb. (KgCO ₂)	Emp. Carb. (KgCO ₂)	Emp. Carb. (KgCO ₂)	Emp. Carb. (KgCO ₂)	Emp. Carb. (KgCO ₂)
1	1, 11, 6, 18, 60, 34	4270.75	4270.75	4878,72	4495,35	4730,21
2	1, 11, 6, 30, 60, 34	3227.42	3227.42	3331,22	3258,16	3286,66
3	1, 11, 6, 30, 76, 31	5043.47	5050.76	6378,43	5631,13	5744,03
4	1, 15, 10, 30, 60, 34	5050.87	5107.04	5610,38	5383,75	5509,85
5	1, 15, 10, 50, 60, 34	3217.98	3297.38	3468,50	3381,18	3324,33
6	1, 9, 4, 12, 56, 24	11131.66	11172.14	12021,73	11697,84	11875,23
7	1, 9, 4, 16, 56, 24	9377.61	10385.73	10268,81	9728,49	10159,73
8	2, 22, 12, 60, 60, 34	2972.66	3030.81	3078,19	3040,02	3046,37
9	2, 30, 20, 100, 60, 34	2740.07	2810.46	3026,48	3015,67	2839,86
10	2, 18, 8, 24, 60, 30	12328.54	12355.2	13222,04	12425,35	12765,57
11	2, 22, 12, 36, 46, 32	2937.75	3464.57	3243,49	3106,44	3174,49
12	2, 22, 12, 36, 52, 32	3050.41	3087.07	3110,37	3080,23	3129,30
13	2, 22, 12, 60, 46, 32	2742.17	2785.63	2817,70	2812,07	2802,77
14	2, 22, 12, 60, 52, 32	2917.44	2952.33	3141,47	3102,45	3113,42
15	3, 33, 18, 54, 60, 34	2566.64	2613.54	2637,95	2624,83	2630,27
16	3, 33, 18, 54, 46, 32	2437.97	2437.97	2695,55	2561,12	2443,96
17	3, 33, 18, 54, 52, 32	2641.56	2655.14	2757,30	2744,18	2761,15
18	3, 33, 18, 90, 46, 32	2437.97	2437.97	2676,27	2665,34	2597,21
19	3, 33, 18, 90, 52, 32	2641.56	2644.30	3005,42	2995,47	2785,73
20	4, 44, 24, 72, 51, 45	1953.49	2028.90	2083,75	2047,89	2147,69

Ce tableau démontre que pour les petites instances, l'algorithme mémétique est capable d'obtenir les plus bas coûts d'empreinte carbone pour 70% des instances utilisées. Il est suivi par le recuit simulé, qui dans certains cas, a pu obtenir des résultats meilleurs que le MA. En effet, pour les instances 5, 9, 13, 16, 18 et 19 l'empreinte carbone est moins élevée que les deux autres algorithmes à savoir l'algorithme génétique et l'algorithme mémétique. L'algorithme génétique, quant à lui, présente les moins bons résultats.

À partir de ce même tableau, nous remarquons que contrairement à l'algorithme de recherche tabou itéré, les trois algorithmes utilisés n'ont réussi, dans aucun cas, à obtenir la même valeur d'empreinte carbone obtenue par la méthode exacte. Même si pour certaines instances telles que les instances 7, 11 et 12 nous avons réussi avec MA et SA à obtenir de meilleurs résultats que l'ITS.

On pourrait conclure que pour les instances de petite taille, l'algorithme de recherche tabou itéré s'avère le plus efficace.

Afin d'évaluer les performances des algorithmes utilisés pour les problèmes de grande taille, nous les avons également comparés à l'algorithme de recherche tabou itéré.

Le tableau 4.9 démontre que pour les instances de grande taille, nous avons réussi à avoir de bons résultats avec l'algorithme mémétique par rapport à ITS à l'exception des instances 6, 9 et 11. Le recuit simulé a aussi réussi à obtenir des résultats meilleurs que l'ITS pour les instances 1, 2, 3, 4, 5 et 7. L'algorithme mémétique reste, pour les instances à grande taille, l'algorithme qui donne les meilleurs résultats comparés à ITS, GA et SA.

Tableau 4.9: Comparaison entre ITS, GA, MA et SA pour les instances de grande taille

Inst.	(D, T, X, S, V, F)	ITS	GA	MA	SA
		Emp. Carb. (t)	Emp. Carb. (t)	Emp. Carb. (t)	Emp. Carb. (t)
1	1, 15, 10, 50, 430, 931	25,31	24,68	21,41	21,41
2	1, 15, 10, 50, 550, 1151	31,34	33,16	29,48	29,38
3	2, 30, 20, 100, 500, 929	16,91	17,44	15,21	15,41
4	2, 30, 20, 100, 500, 1021	15,37	16,42	14,4	14,98
5	4, 60, 40, 200, 580, 1057	11,84	11,18	10,04	10,36
6	6, 90, 60, 300, 560, 817	9,29	10,45	9,8	10,09
7	6, 90, 60, 300, 770, 1189	15,3	15,64	14,24	14,78
8	8, 120, 80, 400, 660, 805	12,9	13,42	12,76	18,03
9	8, 120, 80, 400, 770, 931	13,7	16,37	15,15	17,42
10	10, 150, 100, 500, 860, 1026	16,17	17,2	16,13	17,16
11	10, 150, 100, 500, 890, 1007	14,63	17,26	16,44	16,88

Pour les trois algorithmes implémentés dans ce travail, l'algorithme mémétique présente les meilleurs résultats aussi bien pour les petites comme pour les grandes instances. Le recuit simulé, lui, permet de donner un coût d'empreinte carbone un peu plus élevé, mais dans des délais plus rapides que celle avec l'algorithme mémétique.

Le temps d'exécution élevé de l'algorithme mémétique par rapport à l'algorithme de recuit simulé s'explique par l'intervention de l'algorithme de recherche tabou sur chaque individu dans chaque génération. Un algorithme TS participe grandement à l'amélioration de la solution sauf que l'exécuter plusieurs fois dans une même itération augmente légèrement le délai de traitement de

l'algorithme mémétique. Le compromis entre le temps d'exécution de l'algorithme ainsi que l'amélioration de la solution obtenue est donc trouvée grâce par MA.

Une fois que les algorithmes utilisés ont été comparés avec d'autres algorithmes nous allons agrandir encore plus les instances pour pouvoir ainsi comprendre le comportement de ces derniers principalement en termes de temps.

De ce fait, nous allons utiliser, pour l'algorithme mémétique avec l'algorithme de recherche tabou, la même architecture et augmenter pour chaque itération le nombre machines virtuelles.

Tableau 4.10 Résultat détaillés du MA pour les instances de grande taille

Inst.	(D, T, X, S, V, F)	<i>Emp. Carb.</i> (<i>t</i>)	Nb serveurs inactifs (%)	Temps d'exécution (s)	Num. itération
1	10, 300, 200, 500, 1000, 1000	35.68	175	1032.36	101
2	10, 300, 200, 500, 2000, 2000	47.96	162	1769.25	196
3	10, 300, 200, 500, 3000, 3000	60.14	135	1689.27	112
4	10, 300, 200, 500, 4000, 4000	74.21	129	3224.43	132
5	10, 300, 200, 500, 5000, 5000	93.67	72	4159.78	279

D'après le tableau 4.10, nous constatons que le temps d'exécution de l'algorithme pour retrouver la meilleure solution, augmente considérablement quand on augmente encore plus la taille des instances. En effet, il faudra prévoir une soixantaine de minutes pour retrouver un bon placement de 5000 machines virtuelles dans un environnement de 10 data centers.

Le nombre de serveurs actifs diminue lorsqu'on augmente le nombre de machines virtuelles. C'est tout à fait normal étant donné que les combinaisons possibles des machines virtuelles augmentent et donc plus de serveurs seront mis à disposition pour satisfaire un bon placement des VMs.

CHAPITRE 5 CONCLUSION

Trouver un bon placement des machines virtuelles dans un environnement infonuagique est une tâche importante pour la réduction de l'empreinte carbone d'un data center. Plusieurs approches ont été élaborées pour la résolution du VMPP. Notre objectif était donc de concevoir une approche heuristique capable de résoudre le problème de placement des machines virtuelles dans l'optique de réduire l'empreinte écologique.

Dans ce chapitre, nous allons faire une synthèse des travaux réalisés. Puis, nous présenterons les limites de nos travaux pour finir avec les travaux futurs.

5.1 Synthèse des travaux réalisés

Pour la résolution de ce problème, nous avons implémenté un modèle de consommation énergétique qui tient compte du besoin en énergie de tous les composants d'un data center, du trafic réseau, ainsi que du système de refroidissement. En combinant tous ces paramètres dans un même modèle, nous avons pu concevoir notre algorithme.

Nous avons tout d'abord implémenté l'algorithme génétique et avons pu obtenir des résultats plutôt satisfaisants. Ensuite, nous avons conçu un algorithme hybride. Ce dernier est l'algorithme mémétique qui consiste en une combinaison de deux métaheuristiques à savoir l'algorithme génétique et un algorithme de recherche local. L'algorithme de recherche local utilisé est l'algorithme de recherche tabou. Avec l'algorithme mémétique, nous avons pu améliorer les résultats obtenus auparavant avec l'algorithme génétique. Cette amélioration a par conséquent affecté le temps d'exécution qui lui, s'est vu augmenter.

Pour l'évaluation de l'algorithme mémétique, nous avons implémenté un dernier algorithme qui est le recuit simulé. Avec ce dernier, nous avons amélioré le temps d'exécution, tandis que l'empreinte carbone qui est l'objectif principal de ce travail, n'a pas été améliorée comparé à l'algorithme mémétique, bien au contraire elle a été légèrement augmentée.

L'évaluation de l'ensemble des algorithmes utilisés dans ce travail a nécessité une comparaison avec la méthode exacte pour les petites instances et l'algorithme de recherche tabou itéré pour les grandes instances.

Les résultats obtenus prouvent l'efficacité de l'algorithme mémétique pour les instances de grande taille. En effet, l'algorithme mémétique a réussi à obtenir les meilleurs coûts d'empreinte carbone. Pour les instances de petite taille, certes, l'algorithme de recherche tabou itéré reste l'algorithme le plus efficace, néanmoins l'algorithme mémétique présente un coût d'empreinte carbone pas loin de celui obtenu avec ITS voire parfois mieux.

Ainsi, dans ce travail, nous avons présenté un algorithme mémétique avec recherche tabou qui, dans un cadre de placement des machines virtuelles dans un environnement infonuagique, est capable de trouver la meilleure organisation de VMs dans un ensemble de data centers géographiquement distants tout en garantissant une bonne qualité de service.

Un fournisseur de service infonuagique pourra adopter l'approche élaborée dans ce travail pour effectuer un bon placement des machines virtuelles de ses clients, sans pour autant dépenser des frais énormes. Cet algorithme entre principalement en jeu lors d'une nouvelle assignation des VMs dans un environnement infonuagique initialement vide.

5.2 Limitation des travaux

Ce travail effectué permet de trouver un bon placement des machines virtuelles dans un environnement infonuagique grâce à un algorithme de métaheuristique hybride. En dépit des nombreux avantages que peut apporter cette approche comme contributions aux fournisseurs de services cloud, plusieurs limitations persistent encore dans cette dernière :

En effet, dans ce travail nous ne traitons pas le dynamisme des machines virtuelles après leurs placements puisque les besoins en ressources des VMs peuvent varier au cours de leurs utilisations. Pour remédier à cela, nous prévoyons une certaine marge libre de la totalité des ressources d'un serveur lors de l'allocation des ressources aux VMs, ce qui rend l'étape de placement imprécise.

De plus, lors de l'assignation d'une machine virtuelle à un serveur, nous ne prenons pas en considération son historique afin de prévoir son besoin en termes de ressources dans le temps.

Aussi, nous ne considérons pas que des machines virtuelles complémentaires puissent communiquer entre elles en étant dans des data centers différents. Cette liaison inter-data center, considérée comme coûteuse dans notre approche, pourrait être dans certains cas une contrainte imposée par le client. Nous devrions donc étudier toutes les possibilités de trafic réseau dans l'environnement infonuagique.

5.3 Travaux futurs

Afin d'améliorer ce travail, nous envisageons quelques travaux futurs. Tout d'abord, la prise en considération dans le modèle mathématique utilisé, toute sorte de communication entre tous les composants constituant l'environnement infonuagique. En effet, le trafic réseau ne se limite pas seulement aux machines virtuelles complémentaires d'un même data center. Il faudra donc reconsidérer tous les trafics possibles intra et inter-data center.

Ensuite, nous pourrions envisager d'intégrer un analyseur de trafic réseau permettant de repérer si la bande passante est saturée. Ainsi, nous planifierons, en fonction du trafic, de faire une migration à chaud de la machine virtuelle causant cette saturation vers un autre serveur en utilisant un autre algorithme de placement.

Enfin, nous pourrions prévoir d'autres sortes d'hybridation de l'algorithme mémétique. En effet, essayer un nouvel algorithme de recherche local avec l'algorithme génétique pourrait permettre d'avoir une meilleure solution pour la résolution du problème de placement des machines virtuelles dans un environnement infonuagique.

BIBLIOGRAPHIE

- [1] Y. Ruan, Z. Cao, and Z. Cui, "Pre-Filter-Copy: Efficient and Self-Adaptive Live Migration of Virtual Machines," *IEEE Systems Journal*, vol. 10, no. 4, pp. 1459-1469, 2016.
- [2] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.
- [3] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM computer communication review*, vol. 39, no. 1, pp. 68-73, 2008.
- [4] J. Daniels, "Server virtualization architecture and implementation," *Crossroads*, vol. 16, no. 1, pp. 8-12, 2009.
- [5] D. C. Verma, "Service level agreements on IP networks," *Proceedings of the IEEE*, vol. 92, no. 9, pp. 1382-1388, 2004.
- [6] S.-Y. Jing, S. Ali, K. She, and Y. Zhong, "State-of-the-art research study for green cloud computing," *The Journal of Supercomputing*, pp. 1-24, 2013.
- [7] Y. Feng, B. Li, and B. Li, "Postcard: Minimizing costs on inter-data center traffic with store-and-forward," in *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, 2012, pp. 43-50: IEEE.
- [8] S. Razzaghzadeh, A. H. Navin, A. M. Rahmani, and M. Hosseinzadeh, "Probabilistic Modeling to Achieve Load balancing in Expert Clouds," *Ad Hoc Networks*, 2017.
- [9] S.-W. Ham, M.-H. Kim, B.-N. Choi, and J.-W. Jeong, "Simplified server model to simulate data center cooling energy consumption," *Energy and Buildings*, vol. 86, pp. 328-339, 2015.
- [10] J. Liu, Y. Li, and D. Jin, "SDN-based live VM migration across data centers," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 583-584, 2015.
- [11] Y. Nagafuchi, Y. Teramoto, B. Hu, T. Kishi, T. Koyama, and H. Kitazume, "Routing optimization for live VM migration between data centers," in *Information and*

- Telecommunication Technologies (APSITT), 2015 10th Asia-Pacific Symposium on*, 2015, pp. 1-3: IEEE.
- [12] M. Callau-Zori, L. Samoila, A.-C. Orgerie, and G. Pierre, "An experiment-driven energy consumption model for virtual machine management systems," IRISA; Université de Rennes 1; CNRS, 2016.
 - [13] S. F. Piraghaj, R. N. Calheiros, J. Chan, A. V. Dastjerdi, and R. Buyya, "Virtual machine customization and task mapping architecture for efficient allocation of cloud data center resources," *The Computer Journal*, vol. 59, no. 2, pp. 208-224, 2016.
 - [14] D. Bouley, "Estimating a data center's electrical carbon footprint," *white paper*, vol. 66, 2010.
 - [15] P. Svärd, B. Hudzia, J. Tordsson, and E. Elmroth, "Evaluation of delta compression techniques for efficient live migration of large virtual machines," *ACM Sigplan Notices*, vol. 46, no. 7, pp. 111-120, 2011.
 - [16] X. Li, J. Wu, S. Tang, and S. Lu, "Let's stay together: Towards traffic aware virtual machine placement in data centers," in *INFOCOM, 2014 Proceedings IEEE*, 2014, pp. 1842-1850: IEEE.
 - [17] A. Aggarwal and S. Malhotra, "Goals and Constraints for Devising Efficient Heuristics in Virtual Machine Placement Plan," in *Advances in Computing and Communication Engineering (ICACCE), 2015 Second International Conference on*, 2015, pp. 98-103: IEEE.
 - [18] D. Perez-Palacin, R. Mirandola, F. Monterisi, and A. Montoli, "QoS-driven probabilistic runtime evaluations of virtual machine placement on hosts," in *Utility and Cloud Computing (UCC), 2015 IEEE/ACM 8th International Conference on*, 2015, pp. 90-94: IEEE.
 - [19] L. Zhang, X. Yin, Z. Li, and C. Wu, "Hierarchical virtual machine placement in modular data centers," in *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, 2015, pp. 171-178: IEEE.

- [20] T. Fukunaga, S. Hirahara, and H. Yoshikawa, "Virtual machine placement for minimizing connection cost in data center networks," in *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, 2015, pp. 486-491: IEEE.
- [21] Z. Zhang, C.-C. Hsu, and M. Chang, "Cool cloud: a practical dynamic virtual machine placement framework for energy aware data centers," in *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, 2015, pp. 758-765: IEEE.
- [22] L. Li, K. Liu, B. Fu, M. Chen, and L. Zhang, "Guarantee-aware cost effective virtual machine placement algorithm for the cloud," in *Proceedings of the ACM International Conference on Computing Frontiers*, 2016, pp. 347-348: ACM.
- [23] J. Zhang, Z. He, H. Huang, X. Wang, C. Gu, and L. Zhang, "SLA aware cost efficient virtual machines placement in cloud computing," in *Performance Computing and Communications Conference (IPCCC), 2014 IEEE International*, 2014, pp. 1-8: IEEE.
- [24] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Transactions on Cloud Computing*, vol. 3, no. 1, pp. 14-27, 2015.
- [25] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Physical machine resource management in clouds: A mechanism design approach," *IEEE Transactions on Cloud Computing*, vol. 3, no. 3, pp. 247-260, 2015.
- [26] J. Dong, X. Jin, H. Wang, Y. Li, P. Zhang, and S. Cheng, "Energy-saving virtual machine placement in cloud data centers," in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, 2013, pp. 618-624: IEEE.
- [27] L. He, "A method of virtual machine placement based on gray correlation degree," in *Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on*, 2014, pp. 419-424: IEEE.
- [28] H. Deng, L. Huang, C. Yang, H. Xu, and B. Leng, "Minimizing response latency via efficient virtual machine placement in cloud systems," in *Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance*, 2015, pp. 1-8: IEEE.
- [29] V. D. Justafort, R. Beaubrun, and S. Pierre, "On the carbon footprint optimization in an intercloud environment," *IEEE Transactions on Cloud Computing*, 2015.

- [30] V. Justafort, R. Beaubrun, and S. Pierre, "A Hybrid Approach for Optimizing Carbon Footprint in InterCloud Environment," *IEEE Transactions on Services Computing*, 2016.